

METODOLOGÍA PARA OPTIMIZACIÓN, A TRAVÉS DE ALGORITMOS GENÉTICOS, DE SISTEMAS DE MANUFACTURA FLEXIBLE LIBRES DE BLOQUEOS MODELADOS MEDIANTE REDES DE PETRI

Cristian Camilo Grisales Londoño
David Daniel Villegas Trujillo

METODOLOGÍA PARA OPTIMIZACIÓN, A TRAVÉS DE ALGORITMOS GENÉTICOS, DE SISTEMAS DE MANUFACTURA FLEXIBLE LIBRES DE BLOQUEOS MODELADOS MEDIANTE REDES DE PETRI

Cristian Camilo Grisales Londoño
David Daniel Villegas Trujillo

Trabajo de grado presentado como requisito
parcial para optar al título de
Ingeniero Electricista

Pereira, Octubre de 2017
UNIVERSIDAD TECNOLÓGICA DE PEREIRA
Programa de Ingeniería Eléctrica.



METODOLOGÍA PARA OPTIMIZACIÓN, A TRAVÉS DE ALGORITMOS GENÉTICOS, DE
SISTEMAS DE MANUFACTURA FLEXIBLE LIBRES DE BLOQUEOS MODELADOS MEDIANTE
REDES DE PETRI

Cristian Camilo Grisales Londoño

David Daniel Villegas Trujillo

Director: Mauricio Holguín Londoño

Pereira, Octubre de 2017

Programa de Ingeniería Eléctrica.

Universidad Tecnológica de Pereira

Agradecimientos

Al Ing. Mauricio Holguín por la guía y el espacio que nos brindó
A mi familia por el apoyo durante mi carrera
A los compañeros que aportaron a este proyecto
A la Universidad por todos sus aportes a mi formación
Cristian Camilo Grisales Londoño

Al Ing. Mauricio Holguín por la paciencia que nos tuvo
A mi familia por estar siempre a mi lado
A mis amigos por ese soporte continuo
A mi Abuelo que me está observando
David Daniel Villegas Trujillo

Resumen

Dada la amplia gama que se puede dar en los Sistemas de Manufactura Flexible (SMF) y lo tedioso que puede llegar a ser el seguimiento de los diferentes procesos, es de vital importancia encontrar una metodología de representación de estos procesos. El concepto de fabricación flexible posee un desarrollo simultáneo con la tecnología informática en el ámbito del diseño y la fabricación de piezas o productos, con la ayuda de las Redes de Petri es posible lograr la representación gráfica y sencilla de interpretar, ya que estas se han utilizado con éxito en la literatura para representar automatismos y su modelado es sencillo y fácil de interpretar.

Las Redes de Petri propensas a tener bloqueos son uno de los principales inconvenientes de un SMF, puesto que son funcionamientos indeseados que se deben de evitar o prevenir para tener el mejor desempeño de los recursos del sistema.

En este documento se hace una revisión de los conceptos que abarcan los Sistemas de Manufactura Flexible, Redes de Petri y Algoritmos Genéticos, y luego se presenta una metodología de diseño de Redes de Petri libres de bloqueos además de implementar una estructura para optimizar estas a partir de técnicas metaheurísticas.

Índice general

I	Introducción	1
1.	Planteamiento del problema	2
2.	Justificación	4
3.	Objetivos	6
3.1.	General	6
3.2.	Específicos	6
II	Materiales y métodos	7
4.	Sistemas de manufactura flexible	8
5.	Redes de Petri	14
5.1.	Propiedades de las RdP	16
5.2.	Matriz de incidencia	17
5.3.	Ecuación de estados	17
6.	Bloqueos	18
7.	Algoritmos genéticos	20
7.1.	Individuos y analogía con algoritmos de búsqueda	21
7.2.	Operadores genéticos	22
III	Metodología	23
8.	Metodología de modelado de SMF por RdP	24
8.1.	Topologías	24

8.2. Metodología	27
8.3. Diseños básicos de SMF representados por RdP	28
9. Metodología de optimización por AG de SMF	34
9.1. Optimización de secuencia de disparos a través de AG	34
9.2. Identificación de rutas a partir de la matriz de incidencia	37
10.Integración entre optimización de disparos y algoritmo de identificación de rutas	41
11.Resultados	44
12.Conclusiones	59

Parte I

Introducción

Capítulo 1

Planteamiento del problema

Dada la alta demanda en la producción de una planta y la búsqueda de la reducción de costos, los sistemas de manufactura flexible (SMF) tienen como objetivo ofrecer una respuesta rápida y económica frente a los cambios que se presentan en su ambiente operativo. Este tipo de manufactura proporciona una consideración especial para diversificar de manera sencilla y eficaz los productos de un lote de tamaño mediano o pequeño para aumentar la competitividad en el mercado a nivel mundial. [2].

Las redes de Petri al ser una herramienta gráfica y matemática, permite visualizar de forma amplia y puntual los sucesos a través del sistema modelado, por ende, las Redes de Petri (RdP) se convierten en esencia en la herramienta predilecta a la hora de modelar SMF ya que representan eventos discretos con sincronizaciones describiendo las relaciones existentes de causa y efecto entre los eventos [3].

Los algoritmos de encaminamiento son una serie ordenada de instrucciones, pasos o procesos que tienen como fin encontrar el camino entre dos puntos en una red [4]. Las rutas en los SMF son los diferentes caminos que puede tomar un diferente proceso hasta concluir su ciclo.

La utilización de algoritmos de encaminamiento es la mejor alternativa si se desea automatizar la selección de rutas, pero este proceso puede generar situaciones indeseadas como los bloqueos. Un bloqueo en un sistema es una interrupción permanente al procedimiento de asignación y liberación de un recurso. Los bloqueos en el encaminamiento de mensajes u objetos emergen como consecuencia de una mala selección de canales o segmentos, que son compartidos por diversos procesos de forma concurrente para mover los mensajes u objetos desde un origen hasta un destino en la red o plataforma de trabajo [3].

En algunas referencias como en [2] los bloqueos se denominan puntos muertos, estos son los problemas de mayor relevancia en el diseño y la etapa de control puesto que pueden repercutir de forma negativa en los SMF como retrasar el proceso o detener la operación completa del sistema.

En [5] y [6] se tratan problemas recurrentes en las redes de Petri, como es el caso de los métodos de desarrollo y solución de los modelos limitados a redes con sus propias especificaciones, puesto que al automatizar solo se puede usar un tipo de red y no varias, lo que esto quiere decir es que un automatismo va a estar asociado a un tipo de red. El otro inconveniente es que este tipo de modelado consume recursos computacionales considerables y en algunos casos no se puede encontrar una posible solución al problema.

En [7] la problemática se enfoca en el diseño de la red donde entre más compleja sea la red menores rutas se tendrán para su desarrollo, y entre más numerosas sean sus rutas, menos compleja será su red.

De lo anterior surge la siguiente pregunta de investigación:

¿Será posible implementar un automatismo empleando como herramienta de modelado las redes de Petri, permitiendo ser libre de bloqueos?

Capítulo 2

Justificación

Actualmente la competitividad es cada vez más alta, por esto es necesario mejorar los sistemas de producción. Colombia presentando una condición netamente agrícola y que no ha encontrado un avance estable a una condición industrial necesita dar pasos agigantados en el ámbito de la automatización para entrar fuertemente a competir con las economías internacionales [8]. En el contexto competitivo demarcado por el cambio técnico innovador resulta imperativo modificar la estructura y composición actual del capital en la industria colombiana para agilizar y facilitar el desarrollo tecnológico mediante investigaciones en tecnología y énfasis en emprendimiento de proyectos de innovación y automatización [9].

Los SMF son una forma de alcanzar dichos estándares, puesto que permiten una mejor administración de espacio, tiempo y recursos. Para el modelamiento de estos automatismos es necesaria una herramienta como las redes de Petri, sin embargo, la implementación de estos sistemas requiere de un alto costo en su inversión inicial, también se debe recuperar los costos de inversión con un volumen de producción elevado. Al modelar los SMF con Redes de Petri la problemática radica en que la red sea lo demasiado robusta para que el sistema siempre tenga continuidad ante cualquier perturbación que se encuentre en esta (Puntos muertos o bloqueos).

Los problemas con respecto a los modelos de los SMF se han abordado usando redes de Petri como modelo, y contemplando diferentes metodologías para darle solución a los bloqueos y que su representación sea la adecuada. Lo que se desea implementar es una metodología que se adapte fácilmente a las redes de Petri evitando los puntos muertos y a su vez el problema tenga la solución más óptima.

Un algoritmo genético (AG) es un método de búsqueda que sigue la teoría de la evolución de Charles Darwin para darle solución a distintos problemas. Para ello, se parte de una población

inicial de la cual se seleccionan los individuos más sobresalientes para luego reproducirlos y mutarlos para finalmente obtener una nueva generación de individuos que serán más aptos que la generación anterior [10]. Los AG se han utilizado ampliamente por su capacidad para brindar buenas soluciones a muchos de los problemas típicos de optimización y también se han desempeñado en áreas como programación automática, aprendizaje de máquina, economía, sistemas inmunes, ecología, genética de poblaciones, entre otras [11].

La programación para una producción óptima de los sistemas de manufactura puede ser muy compleja o imposible de encontrar en tiempos estipulados para dichos sistemas. Esta dificultad se hace presente en el caso de los SMF, donde se reconoce en la literatura que es un problema combinatorio que se clasifica como tipo NP, lo cual quiere decir que no se puede aplicar un algoritmo polinómico para comprobar una solución. Esta característica lleva un método de resolución no determinístico el cual consiste en aplicar métodos heurísticos para obtener soluciones que se van destinando o aceptando a ritmo polinómico, como los Algoritmos Genéticos [5].

El inconveniente que se genera a la hora de utilizar AG es que es un método metaheurístico, por lo tanto, no hay un resultado exacto; pero con la implementación de esta metodología se puede llegar a un resultado óptimo puesto que para este tipo de problema los AG tienen la ventaja de ser conceptualmente simples, de amplia aplicación, robusto ante cambios dinámicos y pueden resolver problemas sin soluciones conocidas [5].

Las RdP permiten una representación compacta y completa de un SMF, y los AG se pueden emplear en conjunto con estos modelos para generar programas cuya funcionalidad sea cercana a la óptima, es así que se combinarán las ventajas de ambos métodos para darle solución a los problemas de programación de las RdP con enfoque especial en puntos muertos para SMF.

Capítulo 3

Objetivos

3.1. General

Presentar una metodología para la programación de automatismos enfocados en sistemas de manufactura flexible, mediante descripción por Redes de Petri libres de bloqueos, y optimización a través de Algoritmos Genéticos.

3.2. Específicos

- Indagar las teorías relevantes de modelado que permitan la descripción de automatismos a través de RdP y también las teorías de optimización por Algoritmos Genéticos, aplicados al problema específico de SMF.
- Determinar una metodología de modelado de SMF mediante RdP, incluyendo la descripción del problema de puntos muertos en sistemas flexibles.
- Presentar una metodología de optimización a través de Algoritmos Genéticos para encontrar SMF modelados por RdP que sean libres de bloqueos.
- Implementar un algoritmo que integre la descripción de SMF por RdP y su optimización libre de bloqueos mediante AG.
- Validar los resultados al comparar con resultados de la literatura.

Parte II

Materiales y métodos

Capítulo 4

Sistemas de manufactura flexible

Los Sistemas de Manufactura Flexibles se componen por varias estaciones de trabajo conectadas por una alimentación de materiales que es capaz de permitir el flujo de trabajos pasando por diversas rutas a través del sistema. Estas estaciones de trabajo se encuentran definidas con un conjunto de máquinas e instalaciones de producción (celdas de maquinado), las cuales funcionan de forma integrada y controlada bajo sistemas computacionales (algunas de ellas por control numérico computacional o CNC por sus siglas) y que tienen la finalidad de una producción de partes o productos dentro de un rango de estilos, tamaños y procesamientos [2] [3].

Sin embargo, para que un sistema de manufactura sea catalogado como flexible debe cumplir diferentes requerimientos, debe tener la capacidad de hacer seguimiento del producto y a su vez distinguir entre los diferentes estilos de productos que se procesan por celda, debe tener la capacidad de realizar un rápido cambio de instrucciones de operación y finalmente permitir cambiar la configuración física con el fin de adquirir nuevas conexiones de trabajo [3].

Ningún sistema de manufactura puede ser completamente flexible puesto que existen límites sobre a cuanta flexibilidad puede ser incorporada, no existe forma de producir una gama infinita de piezas o productos, es por esto que un SMF está diseñado para producir piezas o productos para una gama que contempla estilos, tamaños y procesos, es decir, un SMF tiene la capacidad de producir una familia de una sola pieza o una gama limitada de piezas [4].

Al implementar un SMF independientemente del proceso de fabricación, se desea alcanzar los siguientes objetivos tal y como se menciona en [5]:

- Reducción en los costos de fabricación al eliminar operaciones innecesarias.
- Incremento elevado en los indicadores de productividad, reflejados esencialmente en los volúmenes de fabricación.

- Alcanzar un nivel de calidad significativo para el producto terminado.
- Reducción significativa de espacio de trabajo necesaria para la operación de equipo y maquinaria.
- Llevar al sistema al método JIT (Just In Time o método Toyota) el cual refleja la eliminación de áreas de almacenamiento del producto terminado o de material de producción.

Dependiendo de características tales como el tipo de producto, disposición de espacio y/o economía, se establece un diseño básico de SMF en los siguientes cinco tipos:

- **En línea (In-line):** Tal y como se muestra en la figura 4.1a es una disposición que se presenta como un sistema de transferencia en línea, generalmente capaz de realizar movimientos bidireccionales. En la figura 4.1b se puede apreciar esta disposición en un lavadero automatizado de vehículos, el cual consta de inicialmente un semáforo y una barrera que permien indicar y proporcinar el ingreso al lavadero y circular por las estaciones de trabajo (puesto de mojado, detergente, cepillado y secado) para finalmente salir. Dependiendo de como sea la configuración del sistema se puede prestar o no todos los servicios.

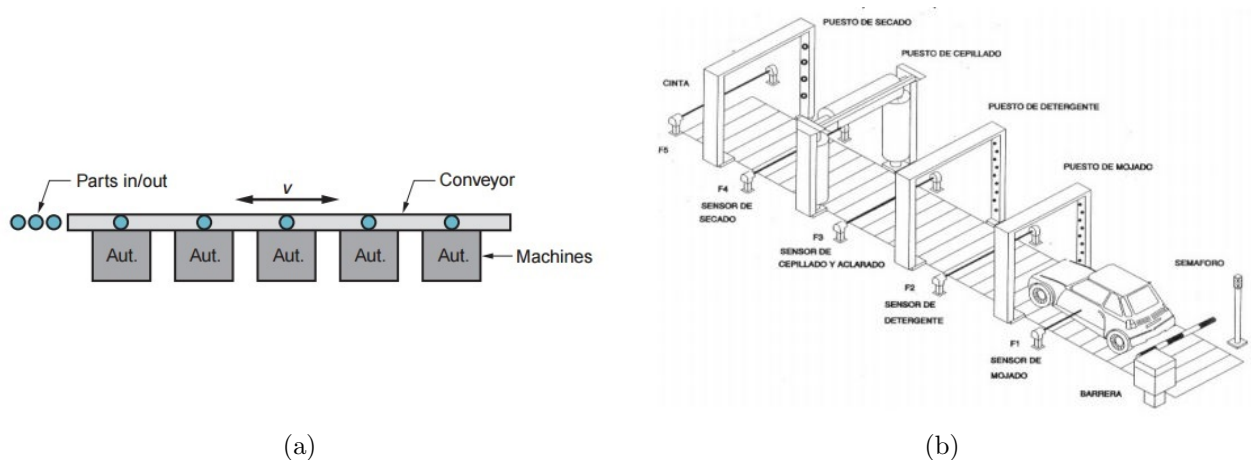


Figura 4.1: (a) Disposición en línea. Fuente: [12] (b) Lavadero automatizado de vehículos

- **Disposición en lazo o bucle (Loop):** Consiste en un diseño de bucle con las estaciones automatizadas alrededor de una banda transportadora; este diseño permite cualquier secuencia de procesamiento dado que cualquier estación es accesible desde cualquier otra estación como se puede ver en la figura 4.2a. La figura 4.2b es la representación de un SMF de tipo lazo que consta de una plataforma giratoria y tres estaciones, entrada del producto (Alimentador), estación automatizada (taladro) y salida del producto (cilindro de evacuación y provador), sin embargo, entre las estaciones de entrada y salida pueden existir tantas estaciones de trabajo como sean necesarias, y su flexibilidad radica en como esté programado el ciclo de trabajo y como se distribuyen las estaciones automatizadas.

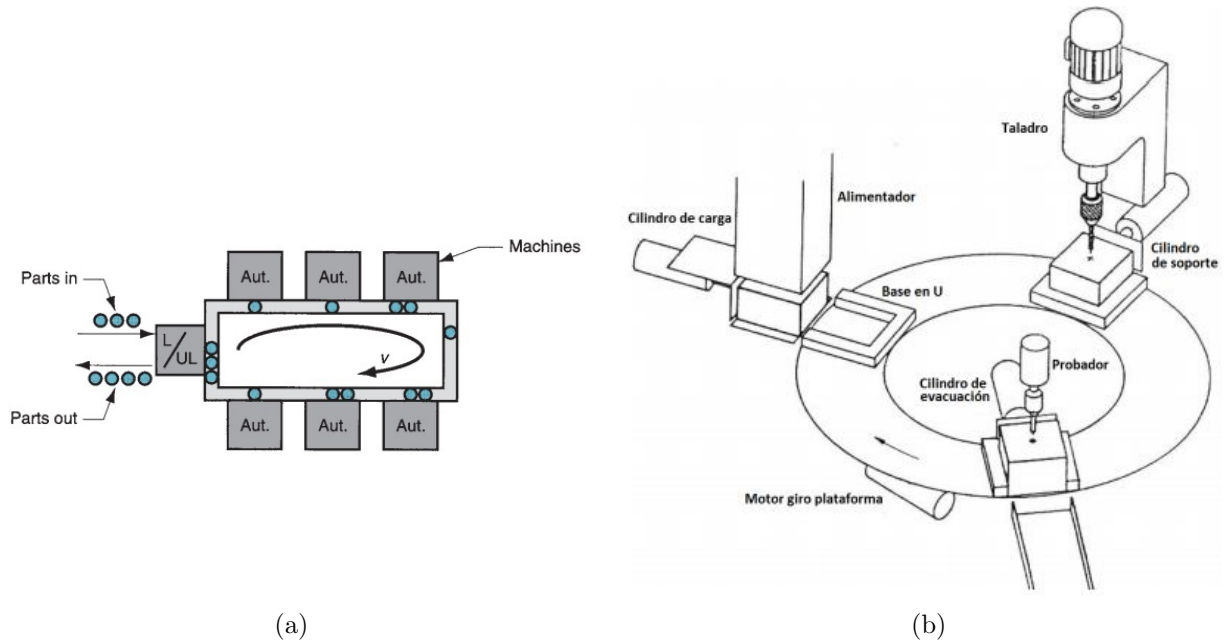


Figura 4.2: (a) Disposición en bucle. Fuente: [12] (b) Banco giratorio

- **Disposición en escalera (Ladder):** Es un diseño en el que se disponen las estaciones automatizadas en peldaños como se puede ver en la figura 4.3a, permitiendo obtener las mismas propiedades de la disposición en lazo. En la figura 4.3b se observa un sistema flexible de atención vehicular de configuración escalera en donde el flujo de éste funciona dependiendo de las necesidades del sistema, y que ha dichas necesidades el vehículo (producto) debe ir a ciertas estaciones.

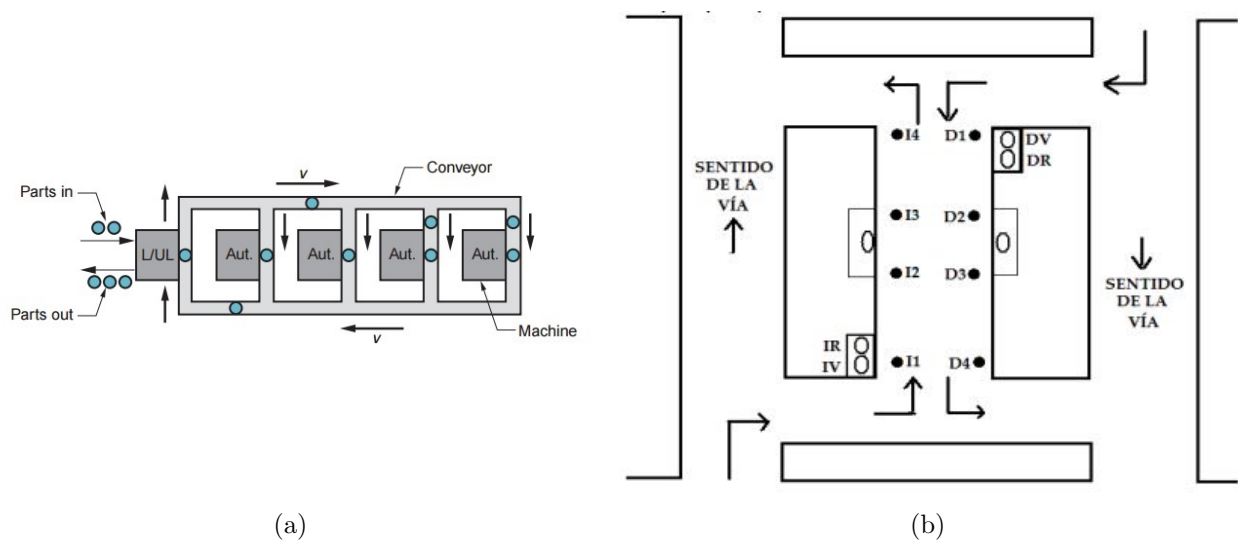


Figura 4.3: (a) Disposición en escalera. Fuente: [12] (b) Sistema flexible de atención vehicular con una disposición en escalera.

- **Disposición de campo abierto (Open Field):** Es la configuración mas compleja que se puede presentar en un SMF, puesto que agrupa varios ciclos de lazo juntos, como se aprecia en la figura 4.4a, además de que el producto llega a cada estación automatizada por medio de vehículos guiados automáticamente (AGV de sus siglas en inglés). La figura 4.4b es una representación mas completa de como sería un SMF con esta disposición, desde las estaciones de carga y descarga, pasando por todas las posibles rutas que puede recorrer el vehículo guiado, por todas las estaciones automatizadas, modulos de inspección, centro de control, entre otras. Esta estructura entrega una percepción mas amplia de las componentes que pueden hacer parte de un SMF de tal magnitud.

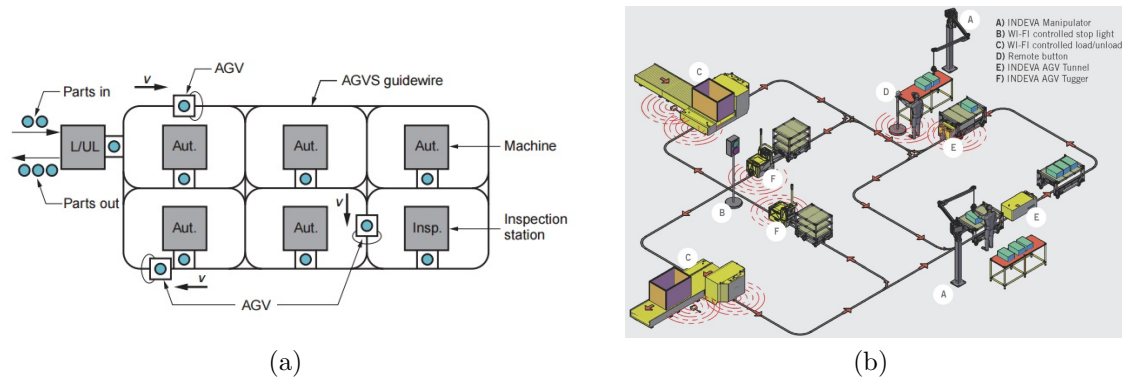


Figura 4.4: (a) Disposición en campo abierto. Fuente: [12] (b) SMF con disposición en campo abierto y todos sus componentes. Fuente: [13]

- **Disposición de celda de Robot (Robotcentered cell):** Presenta un robot en medio de una configuración cuya función y volumen de trabajo se enfatiza en la carga y descarga de producto en estaciones automatizadas. La figura 4.5a es una representación típica de una celda robótica y en la figura 4.5b se aprecia un modelo de esta disposición.

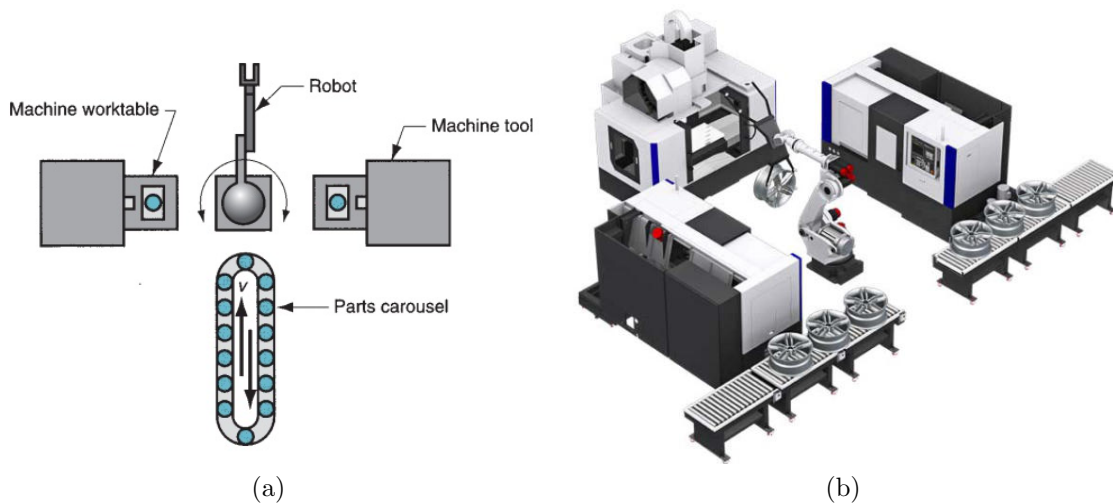


Figura 4.5: (a) Disposición de celda de Robot. Fuente: [12] (b) SMF referente a una disposición de celda de Robot y sus componentes. Fuente: [14]

Dado que un SMF dispone de muchos componentes que se empeñan en realizar diferentes trabajos, se debe de implementar un software que permita integrar todo a un sistema, además de incorporar convenciones que permitan interactuar con todas las máquinas, puesto que ningún SMF se puede automatizar al 100 % ya que en algún punto es indispensable que interceda la labor humana, sea en la carga y descarga de partes del sistema, cambio y ajuste de herramientas, realización de mantenimiento y reparación de equipos, programación y operación del sistema computacional, y en general la administración del sistema. Todas las anteriores son labores que pueden resultar poco viables para automatizar o simplemente no se pueden realizar por un sistema automatico [12].

Algunas de las ventajas que se pueden observar al implementar una producción continua a una producción con lotes son una mejor programación, reducción del proceso de trabajo y menores plazos de fabricación. Sin embargo, muchas veces la complejidad que se presenta en la estructura de algunos SMF pueden presentar algunos casos que ralentizan el proceso de terminación del producto, generando un inadecuado funcionamiento del sistema, como puede ser la falla de una estación de automatismo o la mala programación del ciclo de producción. Los SMF enfocados en aplicaciones de mecanizado, son los que ven reflejados mayores beneficios, alrededor del 75 % para un SMF y un 40 % a un 50 % para operaciones convencionales de lotes en comparación con la implementación de solamente talleres de máquinas [12].

Capítulo 5

Redes de Petri

Las redes de Petri se reconocen como un grafo dirigido, es una herramienta gráfica y matemática fundamental para el modelado de sistemas dinámicos de eventos discretos. Estas redes poseen dos tipos de elementos: Lugares y transiciones. Los lugares y las transiciones son unidas mediante arcos dirigidos, y estos solo deben unir lugares con transiciones o transiciones con lugares. No se deben unir lugares con lugares, ni transiciones con transiciones [15] [16].

Las marcas son puntos negros que se ubican dentro de cada lugar, puede haber una o varias en un mismo lugar y estas son las que determinan que etapa se esta ejecutando de una red [15].

Los arcos dirigidos poseen un número que indica su peso, el cual determina la cantidad de marcas que se consume o deposita en un lugar.

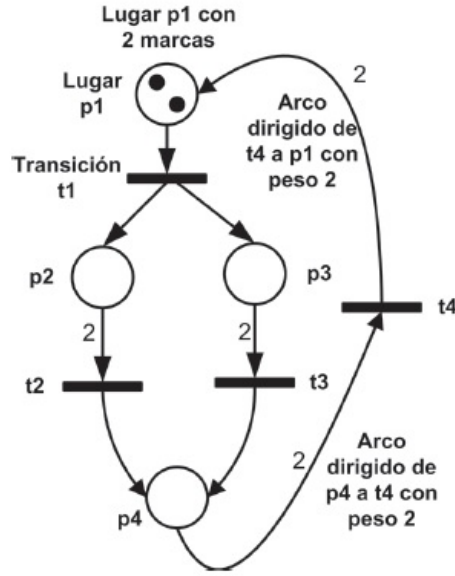


Figura 5.1: Red de Petri. Fuente:[15]

En [16], una red de Petri se define formalmente como una quintupla:

$$RdP = \{P, T, F, W, M_0\}$$

- P se define como un vector determinado por el número de lugares de la red, entonces:

$$P = \{p_1, p_2, p_i, \dots, p_m\}$$

- T se define como un vector determinado por el número de transiciones de la red, entonces:

$$T = \{t_1, t_2, t_i, \dots, t_n\}$$

- F se define como el vector de arcos que determinan el flujo de la red, entonces:

$$F \subseteq (PXT) \cup (TXP)$$

donde: - $\{TXP\}$ Es el conjunto de arcos de salida de transiciones a lugares.

- $\{PXT\}$ Es el conjunto de arcos de entrada de los lugares a transiciones.

- W se define como un vector determinado por el peso de los arcos de F , entonces:

$$W : F \rightarrow \{1, 2, 3, \dots\}$$

- M_0 se define como un vector determinado por el marcado inicial de la red, entonces:

$$M_0 : P \rightarrow \{1, 2, 3, \dots\}$$

La evolución de una RdP hace referencia al movimiento de las marcas en la red, que define el vector de marcaje y los cambios que debe cumplir la regla de disparo de transiciones. Una transición t está habilitada si el número de marcas en los lugares de entrada es mayor o igual al peso del arco que conecta el lugar con la transición. Si una transición está habilitada y su condición lógica vale 1 lógico, entonces se quitan tantas marcas de cada lugar de entrada como peso tiene el arco de entrada a la transición y se ponen tantas marcas en cada lugar de salida de la transición como marcas tiene el arco de salida de la transición.

5.1. Propiedades de las RdP

Hay tres propiedades básicas en las redes de Petri según [16], las cuales son:

1. **Red Viva:** Una RdP es viva, si una vez alcanzado un marcado cualquiera desde M_0 , es posible encontrar una secuencia de disparo para cada una de las transiciones.
2. **Red Límitada:** Para cada lugar existe un límite de marcas. k -límitada, donde k es el número de marcas en cada lugar.
3. **Red Reversible:** Desde todo marcado alcanzable desde M_0 , es posible encontrar una secuencia de disparo que conduzca a M_0 nuevamente.

Otras propiedades o características mencionadas en [16] y [17] son:

- Red binaria = Red segura: Red 1-Límitada.
- Red Conforme = Red binaria + Red viva: GRAFCET.
- Una red es ordinaria si el peso de sus arcos es siempre uno.
- Una red es pura, si no existen auto-bucles, donde los auto-bucles están definidos por una transición t cuya entrada y salida es el mismo lugar p .
- Una transición fuente es aquella que no posee lugar de entrada, y una transición sumidero es aquella que no tiene lugar de salida.

5.2. Matriz de incidencia

En una RdP con n transiciones y m lugares, su matriz de incidencia es una matriz de enteros de $n \times m$, llamada $A = [a_{ij}]$, y a_{ij} está dada por:

$$a_{ij} = a_{ij}^+ - a_{ij}^-$$

donde $a_{ij}^+ = w(i, j)$ es el peso de los arcos que unen la transición i con el lugar de salida j . Está es conocida como la matriz de incidencia posterior. Y $a_{ij}^- = w(i, j)$ es el peso de los arcos que unen el lugar de entrada i hasta la transición j [5].

5.3. Ecuación de estados

La ecuación de estados de una RdP en forma matricial define el estado de la red. Se da un marcado inicial M_0 y una secuencia de disparos de transiciones habilitadas. De tal manera, en [15] se define la ecuación de estados como:

$$M_d = M_o + A^T \sum_{k=1}^d U_k \quad (5.1)$$

donde Md es el vector que define el estado de la red, siendo un vector columna de $m \times 1$. Y U_k es el vector de disparo, siendo un vector columna de $n \times 1$.

Capítulo 6

Bloqueos

Antes de definir qué es un bloqueo, se debe definir qué es un recurso. Ya que estos son los que indican el avance a lo largo de la red.

Un sistema de asignación de recursos (SAR) engloba un conjunto de procesos que tienen un plan predefinido en el que los procesos compiten por los recursos que son finitos. De una forma general, el término recurso se usa en sentido amplio de la palabra y se asocia al medio de cualquier tipo (físico o virtual) que en caso tal, sirve para conseguir lo que se busca [3].

Según las características del comportamiento del sistema, los recursos se pueden agrupar en: consumibles y reutilizables. Los recursos consumibles son aquellos que se destruyen al ser adquiridos por un proceso y un recurso reutilizable es aquel que no se destruye o desgasta con su uso [3].

En este trabajo, las celdas de maquinado se establecen como recursos reutilizables y se representan como plazas en las RdP.

Las rutas de encaminamiento son las encargadas de seleccionar los recursos para llevar un objeto desde un origen hasta un destino. Una incorrecta selección de ruta por parte del algoritmo de encaminamiento puede generar bloqueos, lo cual puede ser algo indeseado para todo el sistema. Un BLOQUEO o PUNTO MUERTO sucede cuando un proceso no puede ser terminado debido a la espera de recursos que son usados por otros procesos. Si durante un proceso ocurre un uso indebido de los recursos existentes, y estos no son liberados oportunamente, puede generar una espera en otros procesos que solicitan los mismos recursos. Si esta espera se hace infinita en el tiempo entre un grupo de procesos donde cada uno requiere un recurso que está siendo utilizado por otro del mismo grupo, se presenta un problema típico de bloqueo [3] [18].

El rendimiento en los sistemas de manufactura se ve afectado debido a la reducción de la utilización de las máquinas, eficiencia en los procesos de producción y retrasos en la terminación de operaciones.

En [18] se mencionan cuatro condiciones que indican que un bloqueo sucede cuando cualquiera de estas esté presente: exclusión mutua, retención y espera, no expropiación y espera circular.

- Exclusión mutua significa que un recurso puede ser utilizado por un sólo proceso, y ningún otro proceso puede utilizarlo a la vez.
- La condición de retención y espera ocurre cuando uno es retenido por un proceso que al mismo tiempo usa otro recurso. Esto provoca un problema en la utilización de recursos, ya que un proceso puede solicitar un recurso, pero usarlo sólo al final de su tarea. Esta condición también conduce a una inanición de algunos procesos que esperan a los recursos sin tiempo específico.
- La no expropiación se presenta cuando el sistema no puede liberar los recursos retenidos por un proceso que requiere otros recursos y es difícil asignar los requeridos en ese momento.
- La condición de espera circular existe cuando hay un recurso solicitado por un proceso y éste es retenido por el siguiente proceso en cadena.

Capítulo 7

Algoritmos genéticos

Como concepto se tiene que los AG son: “ Algoritmos de búsqueda basados en los mecanismos de selección natural y genética natural. Combinan la supervivencia de los más compatibles entre las estructuras de cadenas, con una estructura de información ya aleatorizada, intercambiada para construir un algoritmo de búsqueda con algunas de las capacidades de innovación de la evolución humana” [11].

La evolución, tal y como se conoce, es básicamente un método de búsqueda entre un número enorme de posibles soluciones. En la biología, las posibilidades están dadas por un grupo de secuencia genética posibles y las soluciones esperadas, conformadas por organismos capaces de sobrevivir y reproducirse en sus entornos. La evolución se puede ver también como una forma de construir soluciones a problemas complejos con la capacidad de adaptarse a las variaciones que se presentan con los problemas en distintos medios [11].

El buen funcionamiento de un organismo biológico depende de muchos criterios, que varían a medida que el organismo evoluciona, de modo que la evolución se desarrolla continuamente en un conjunto cambiante de posibilidades. Estos son los motivos por los cuales los mecanismos evolutivos son una inspiración para los algoritmos de búsqueda [11].

Las reglas de evolución, independientemente de su complejidad, son simples: las especies evolucionan mediante variaciones aleatorias, ya sea por mutación, recombinación u otras; seguidas por la selección natural, donde el más apto tiende a sobrevivir y reproducirse, propagando así su material genético a posteriores generaciones [11].

7.1. Individuos y analogía con algoritmos de búsqueda

Todos los organismos vivientes están compuestos por una o más células, cada una de las cuales contiene uno o más *cromosomas* (cadenas de ADN). Éstas son las encargadas de transmitir el material genético de una célula a otra. Los cromosomas se pueden dividir, conceptualmente, en *genes*, que son los encargados de codificar una determinada proteína. Los *alelos* son los encargados de determinar los rasgos del individuo, como por ejemplo el color de los ojos. Cada gen está ubicado en una determinada posición o lugar en un cromosoma.

El *genoma* de un organismo es la colección completa del material genético y el *genotipo* es el conjunto de genes contenidos en un genoma. Por *fenotipo* se conoce a las cualidades físicas u observables en un organismo, como la fisiología, la morfología y la conducta [11].

La mayoría de las especies reproductoras sexualmente en la tierra almacenan sus cromosomas por parejas (son diploides). En el caso del ser humano, cada célula somática (no germen) del cuerpo, contiene 23 pares de cromosomas. Durante la reproducción sexual se produce una recombinación o cruce entre los cromosomas de los padres, es decir, se intercambian genes en cada par de cromosomas, para formar un gameto, y así, los gametos de los padres se emparejan para formar un conjunto de cromosomas diploides. En algunos casos se puede dar una reproducción haploide (sus cromosomas no están conformados por parejas, son simples), los genes se intercambian entre dos padres con una sola rama de cromosomas. La descendencia está sujeta a mutaciones, donde se produce una alteración en algún nucleótido (bits elementales del ADN) de padre a hijo [11].

La capacidad de un organismo se define como la probabilidad de que el organismo viva y pueda reproducirse, es la viabilidad, o como función del número de descendencia que tenga dicho organismo, o sea, su fertilidad.

Análogamente en los AG, el término *cromosoma* se refiere al de un individuo que es una posible solución del problema, que se codifica como una cadena de bits. Los *genes* pueden ser un bit o un pequeño conjunto de bits adyacentes que codifican un elemento particular del individuo candidato a solución. Un *alelo* en una cadena de bits puede ser un 0 o un 1. El *cruce* consiste en un intercambio de material genético entre dos cromosomas de dos padres haploides. La *mutación* consiste en el intercambio de un bit en un lugar aleatorio. En la mayoría de aplicaciones de los AG se encuentran individuos haploides [11].

El *genotipo* de un individuo es la configuración de bits del cromosoma al que pertenece éste. El *fenotipo* es la expresión del genotipo, es decir, una cadena de bits [11].

La codificación es en términos generales, una forma de representación de los individuos encontrados.

7.2. Operadores genéticos

■ Selección

Este operador selecciona de la población actual que participa en la configuración de la nueva población. La función de este termina una vez se decida el número de descendientes que debe tener la población actual. La forma más simple de implementar la selección es usando el denominado esquema de selección proporcional. En esta configuración cada individuo tiene el derecho a generar un número de descendientes que es proporcional a la adaptación de cada individuo. Aunque experimentalmente la selección proporcional tiene dos problemas: el primero, es que algunas configuraciones en las primeras fases pueden contaminar nuevas generaciones porque se podrían generar muchos descendientes, produciendo una convergencia prematura a un óptimo local; el segundo problema es que en las fases finales existe el problema de pérdida de selectividad, debido a que casi todas las configuraciones tienen funciones de adaptación muy similares [5].

■ Recombinación

El operador de recombinación (crossover) se denomina como un operador que intercambia el material genético de un par de padres produciendo descendientes que normalmente son diferentes de sus padres; la idea esta centrada en que segmentos distintos de padres diferentes y que poseen alta adaptación deberían combinarse en nuevos individuos que toman ventaja de esta combinación [5][19].

■ Mutación

La mutación, en la codificación binaria, se explica en como cambia el valor de una variable de 0 a 1 ó viceversa. Juega un papel secundario y con tasas de mutación pequeñas garantiza que un individuo no difiera mucho de sus padres, este operador sirve para evitar perder diversidad producida por los bits que convergen a cierto valor para toda la población. Funciona invirtiendo cada bit de lo que se denomina *tirabinaria* sobre una base bit a bit con una probabilidad P_m , siendo P_m la tasa de probabilidad de mutación [19][20].

Parte III

Metodología

Capítulo 8

Metodología de modelado de SMF por RdP

Un sistema de manufactura flexible consiste en una variedad de componentes como robots, máquinas, materia en bruto (considerado como la fase inicial del producto antes de iniciar su procesamiento), sensores, actuadores, computadores y accesorios relacionados a procesos específicos. Pero antes de obtener una RdP equivalente a todo lo anterior, es necesario implementar una metodología que permita entender y modelar el sistema a cierto grado, analizar las características de cada componente del sistema y la relación entre ellos, hacer suposiciones para así lograr eficiencia y simplicidad en el diseño y el análisis. Entonces, para lograr un diseño óptimo es importante tener en cuenta las leyes físicas por las que se rigen los componentes y sus enlaces.

8.1. Topologías

Es necesario un enfoque en las principales operaciones y sus secuencias o precedencias, concurrencias o relaciones conflictivas para la etapa de modelamiento. Estas relaciones básicas entre procesos y operaciones pueden ser clasificadas por los siguientes tipos:

- **Secuencial:** Es el caso en el que una operación es continua a otra, entonces los lugares y/o transiciones se ven representados por una relación secuencial, o en cascada, tal y como se observa en la figura 8.1.

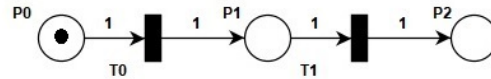


Figura 8.1: Secuencial

- **Concurrente:** cuando dos operaciones son iniciadas por un evento, forman una estructura paralela, en donde dos lugares son salidas de la misma transición. Las operaciones equivalentes a estos canales concurrentes pueden ser representadas con una serie de lugares y transiciones secuencialmente conectados en donde múltiples lugares pueden ser marcados simultáneamente o múltiples transiciones son habilitadas en ciertas marcas. Un ejemplo de este suceso se puede observar en la figura 8.2, donde P0 y P1 deben de presentar una marca para que T0 pueda sensibilizarse y se otorgue marcado en P2 y P3.

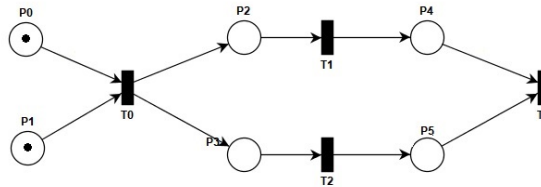


Figura 8.2: Concurrente

- **Conflictivo:** si de una operación representada por un lugar se tiene como salidas dos transiciones que dan continuidad a varias operaciones, entonces dos transiciones tendrán como salida un mismo lugar. Este caso se puede observar en la figura 8.3 donde hay marca en P0 y las transiciones T0 y T1 están sensibilizadas, por lo tanto, puede tomar cualquiera de los dos caminos.

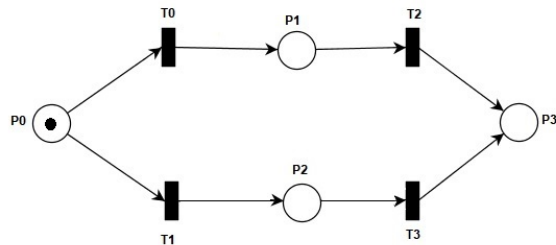


Figura 8.3: Conflictivo

- **Cíclico:** es una secuencia de operaciones continúa una después de otra, donde al finalizar la última inicia la primera, describen una estructura cíclica entre estas operaciones como se observa en la figura Figura 8.3.

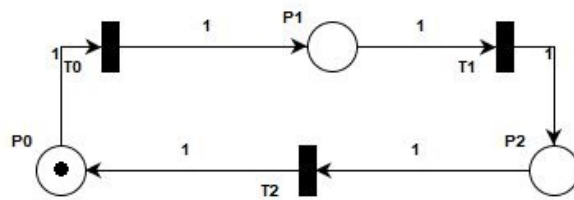


Figura 8.4: Ciclico

- **Exclusión mutua:** dos procesos son mutuamente excluyentes si no pueden ser ejecutados al mismo tiempo debido a restricciones en el uso de recursos compartidos. Para implementar esta condición es necesario disponer de un lugar común marcado como se observa en P1 de la figura 8.5, además de múltiples arcos de salida y entrada para poder activar estos procesos. Un ejemplo puede ser el de un robot el cual es compartido por dos máquinas para las funciones de carga y descarga. La representación de un recurso compartido se puede ver en la Figura 8.5.

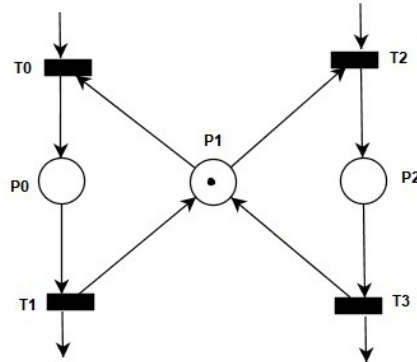


Figura 8.5: Exclusión mutua

Los recursos representados por las marcas en la RdP se pueden clasificar por dos tipos, recursos de marcas dedicadas o recursos de marcas compartidas. Las dedicadas ocupan un lugar con arcos de entradas y salidas sencillas; las compartidas ocupan un lugar con arcos de múltiples entradas y múltiples salidas.

8.2. Metodología

Teniendo claridad sobre la clasificación de marcado y las topologías representativas de las RdP, se puede aplicar entonces una metodología de modelado general resumida en los siguientes pasos:

1. Identificación de Operaciones y Recursos.

Dada la descripción de un sistema automatizado, es primordial identificar los eventos, procesos, recursos y condiciones principales entre otros.

2. Identificación de Relaciones.

La relación entre los elementos anteriormente mencionados debe ser determinada, los recursos, ya sean dedicados o compartidos tienen restricciones que deben ser determinadas para la asignación de una ruta de encaminamiento del mismo.

3. Diseño de la Red de Petri.

De acuerdo a las relaciones identificadas en el paso anterior y según el evento, operación o proceso se deben organizar los lugares y/o transiciones para así poder ser asignadas y etiquetadas. Es necesario introducir lugares y transiciones suficientes para que cumplan con la estructura de una RdP. El número de marcas iniciales y sus posiciones es acorde al estado inicial del sistema y se debe asociar de ser necesario otras características a las etapas o transiciones, tales como tiempo o prioridad.

4. Modificación de la Red de Petri.

Verificar que la red modelo refleje la operación del sistema y modificarla hasta que esta modele el sistema. Algunos sistemas no pueden ser modelados usando RdP ordinarias sin algunas extensiones como arcos inhibidores o prioridades en transiciones.

8.3. Diseños básicos de SMF representados por RdP

En esta sección se discuten algunos ejemplos asociados a los tipos de SMF hablados en el Capítulo 4, donde se hace una breve descripción del sistema y se asocia una posible representación a través de Redes de Petri.

- **SMF Tipo escalera:** El sistema de manufactura consta de 2 estaciones. La primera estación consta de 3 máquinas y la segunda estación de 2 máquina [17]. El trabajo se divide en 2 operaciones, donde la operación 1 se realiza en la estación 1 y la operación 2 se realiza en la estación 2. El esquema del SMF se puede ver en la figura 8.6.

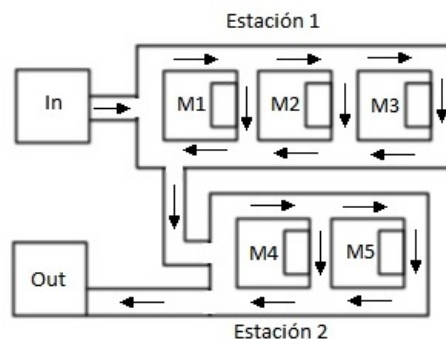


Figura 8.6: SMF tipo escalera

Se puede observar que la figura 8.6 es similar a la topología tipo escalera (figura 4.3). Además, se identifica cada estación como una topología conflictiva (figura 8.3)

y siguiendo los pasos de la metodología de representación de un SMF a RdP, la red resultante es la siguiente:

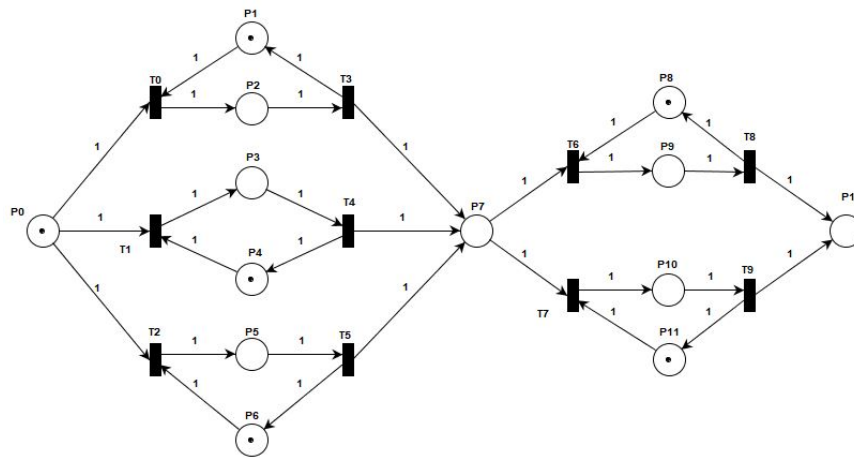


Figura 8.7: RdP del SMf tipo escalera

En la tabla 8.1 se puede ver la descripción de los lugares de la RdP anterior.

Cuadro 8.1: Descripción de los lugares de la RdP 8.7

Lugar	Descripción	Lugar	Descripción
P0	Operación 1 disponible	P7	Operación 1 finalizada
P1	Máquina 1 disponible	P8	Máquina 4 disponible
P2	Operación 1 en proceso por Máquina 1	P9	Operación 2 en proceso por Máquina 4
P3	Operación 1 en proceso por Máquina 2	P10	Operación 2 en proceso por Máquina 5
P4	Máquina 2 disponible	P11	Máquina 5 disponible
P5	Operación 1 en proceso por Máquina 3	P12	Operación 12 finalizada
P6	Máquina 3 disponible		

- **SMF de planta de procesamiento de pan de molde:** Basado en el SMF de la figura 8.8, se realiza el modelamiento en RdP de una línea de procesamiento de pan de molde que se encarga de fabricar tres tipos diferentes de pan (tipo 1, tipo 2 y tipo 3). En esta figura se pueden apreciar dos topologías de SMF diferentes (en línea y cíclico). El sistema cuenta con una máquina que se encarga de verter y mezclar los ingredientes

(P0), una máquina para amasar la mezcla (P1), cuatro máquinas porcionadoras (P2, P6, P7 y P8), tres máquinas encargadas de adicionar un nuevo ingrediente (P3, P4, P5) y dos hornos para el proceso de cocción de los panes (P9 y P10).

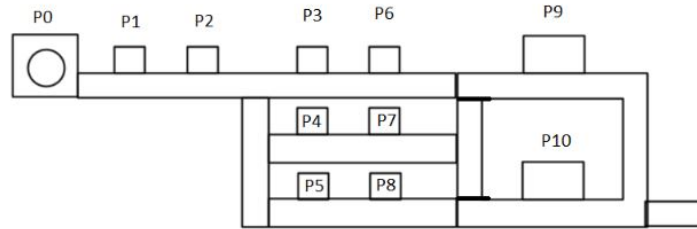


Figura 8.8: SMF de la planta de procesamiento de pan

Siguiendo los pasos de modelamiento para RdP del capítulo 8, se obtiene la siguiente RdP:

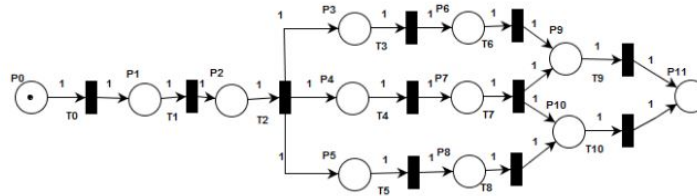


Figura 8.9: RdP de la planta de procesamiento de pan

En la tabla 8.2, se puede ver la descripción de la planta.

Cuadro 8.2: Descripción de los lugares de la RdP de la planta

Lugar	Descripción
P0	Máquina encargada de verter y mezclar
P1	Máquina encargada de amasar
P2	Máquina encargada de porcionar
P3	Máquina encargada de adicionar ingrediente Tipo 1
P4	Máquina encargada de adicionar ingrediente Tipo 2
P5	Máquina encargada de adicionar ingrediente Tipo 3
P6	Máquina encargada de porcionar
P7	Máquina encargada de porcionar
P8	Máquina encargada de porcionar
P9	Horno 1
P10	Horno 2

- **SMF de planta con brazos robóticos:** Este modelo es tomado de [21]. Se tiene un sistema de manufactura el cual consta de 2 trabajos y 3 máquinas (Mach 0, Mach 1 y Mach 2). La ruta de trabajo está dada en la tabla 8.3. Los tiempos se muestran entre paréntesis.

Cuadro 8.3: tabla de trabajos y operaciones

	Trabajo 1		Trabajo 2	
	Máquina	Tiempo	Máquina	Tiempo
Operación 1	Mach 0	2	Mach 0	6
Operación 2	Mach 1 o Mach 5	5	Mach 1	7

Se aprecia que la operación 2 del trabajo 1 puede ser realizado por el Mach 1 o el Mach 2, lo cual representa una estructura conflictiva, el trabajo 1 y el trabajo 2 comparten a Mach 0 y Mach 1, que indica una estructura de recurso compartido.

Nota: La red se modifica con el fin de cumplir los criterios de diseño de RdP para el algoritmo en este documento, los cuales son que se inicie con un solo lugar y termine con un solo lugar. Para los recursos compartidos se tiene en cuenta 2 brazos robóticos

(R1 y R2).

Una vez se sigue la metodología de representación de SMF a RdP, la red representativa es la siguiente:

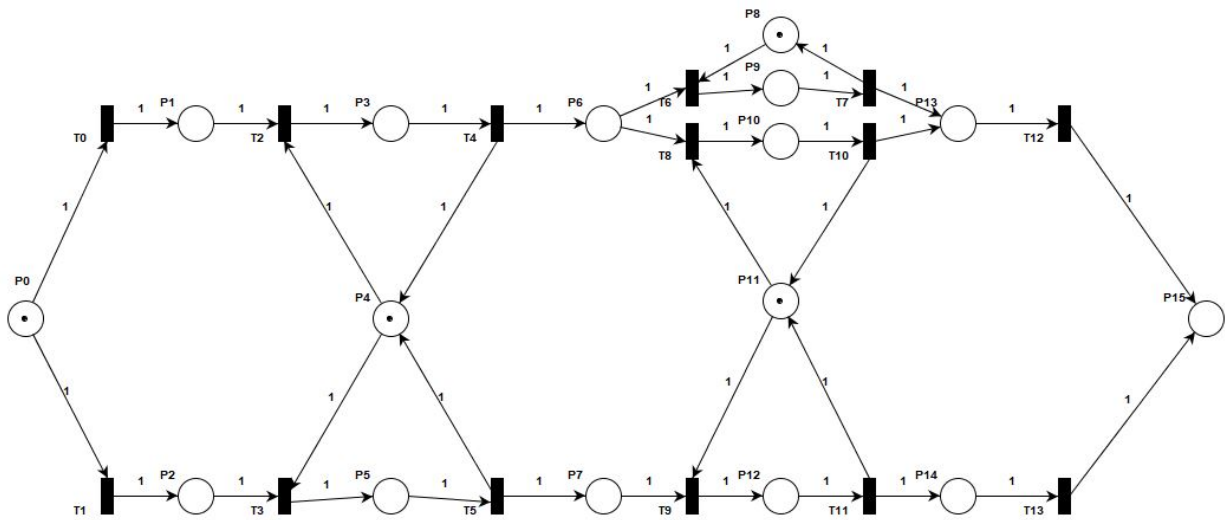


Figura 8.10: RdP SMF de planta con brazos robóticos

En la tabla 8.4, se puede ver la descripción de los procesos.

Cuadro 8.4: Descripción de los lugares y procesos del SMF

Lugar	Descripción	Lugar	Descripción
P0	inicio	P8	Mach 2 disponible
P1	Trabajo 1 disponible	P9	Trabajo 1 en proceso por Mach 2
P2	Trabajo 2 disponible	P10	Trabajo 1 en proceso por Mach 1
P3	Trabajo 1 en proceso por Mach 0	P11	Mach 1 disponible - Robot R2 disponible
P4	Mach 0 disponible - Robot R1 disponible	P12	Trabajo 2 en proceso por Mach 1
P5	Trabajo 2 en proceso por Mach o	P13	Trabajo 1 finaliza operación 2
P6	Trabajo 1 finaliza operación 1	P14	Trabajo 2 finaliza operación 2
P7	Trabajo 2 finaliza operación 1	P15	fin

Capítulo 9

Metodología de optimización por AG de SMF

Teniendo presente una metodología para el diseño de RdP enfocadas en SMF, se hace el uso de AG para encontrar la mejor ruta libre de bloqueos, para esto se utiliza la herramienta de software Matlab© y se implementa una metodología para optimizar la secuencia de disparos.

9.1. Optimización de secuencia de disparos a través de AG

Una secuencia de disparos en una RdP está descrita como las transiciones que fueron sensibilizadas para que se ejecute de forma oportuna, es así que la secuencia de disparos para la Red de Petri de la Figura 9.1 es [T0 T1 T2]

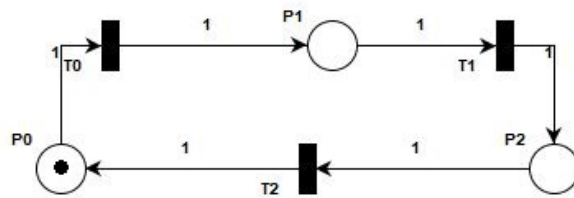


Figura 9.1: Red Ciclica

Para comprobar que las transiciones de la secuencia anterior es válida se utiliza la expresión (5.1) dando como resultado la siguiente secuencia de vectores de estados partiendo del

marcado inicial.

$$\begin{aligned} M0 &= [1 \ 0 \ 0] \\ M1 &= [0 \ 1 \ 0] \\ M2 &= [0 \ 0 \ 1] \\ M3 &= [1 \ 0 \ 0] \end{aligned}$$

Uk , en la expresión (2,1) es el vector de transiciones por lo que se puede hacer un listado de vectores Uk donde se puede visualizar la secuencia de disparos iniciando desde $U0$.

$$\begin{aligned} U0 &= [0 \ 0 \ 0] \\ U1 &= [1 \ 0 \ 0] \\ U2 &= [0 \ 1 \ 0] \\ U3 &= [0 \ 0 \ 1] \end{aligned}$$

Con el fin de encontrar una secuencia de disparos óptima (función objetivo) se hace uso del toolbox de Matlab© **ga** (Genetic Algorithm), con las siguientes características:

- **Codificación:** un modelo planteado requiere de un vector Uk como individuo, el cual está compuesto por una secuencia de disparos. Como ejemplo se tiene el siguiente vector:

$$Uk = [0 \ 1 \ 3 \ 10 \ 5]$$

Donde cada elemento del vector es una variable (cromosoma) que hace referencia a una transición disparada. El valor mínimo y máximo que puede tener cada variable está determinada por el número mínimo y máximo de las transiciones existentes en la RdP a trabajar; entonces la codificación correspondiente al individuo anterior es la siguiente:

$$Uk = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1]$$

- **Selección:** la población inicial se genera automáticamente y de forma aleatoria de 200 individuos, los cuales pasan por un proceso de selección por torneo. Este torneo consiste en que de forma aleatoria se escogen cuatro pares de individuos (padres) y se evalúan en la función objetivo, luego se escogen los individuos (hijos) que tengan mejor adaptabilidad a dicha función y estos individuos se establecen como los nuevos padres para la próxima generación resultante.

- **Recombinación:** una vez obtenidos los resultados de la selección, se escoge el 5 % de los mejores individuos de la población y no se someten al proceso de recombinación. El resto de los individuos pasa por el siguiente proceso:

Se escogen dos padres de forma aleatoria para generar un nuevo individuo (hijo), estos padres se comparan con un vector binario aleatorio del mismo tamaño de los individuos como se muestra en el siguiente ejemplo:

$$U1 = [0 \ 1 \ 3 \ 10 \ 5]$$

$$U2 = [2 \ 7 \ 1 \ 8 \ 9]$$

$$\text{vector binario} = [0 \ 1 \ 1 \ 0 \ 1]$$

Si la variable del vector binario es un 0, se toma el valor de la variable del padre 2 ($U2$), y si la variable del vector binario es un 1, se toma el valor de la variable del padre 1 ($U1$). Una vez terminado este proceso el resultado es:

$$U3 = [2 \ 3 \ 1 \ 8 \ 5]$$

Pseudocódigo de Función Objetivo

Antes de desarrollar el pseudocódigo para la función objetivo del AG es necesario tener la matriz de incidencia, el vector de marcado inicial $M0$, la cantidad de lugares y transiciones. Cabe resaltar que el algoritmo descrito se encuentra limitado a operar con RdP binarias.

Inicio

cargar C //Matriz de Incidencia

cargar $M0$ //Vector de estado inicial

cargar M_{final} //Vector de estado final

cargar Lugares //Número de Lugares

cargar Transiciones //Número de Transiciones

cargar U_k //Vector con la secuencia de disparos expresada en números enteros

$cont \leftarrow 1$

Función $U_k \text{Disparos}(U_k, \text{Transiciones})$ //Transforma U_k en una matriz de disparos

Dimensionar $U_k \text{Matriz}(U_k, \text{Transiciones})$ //Matriz de 0

Desde $i=2$ hasta $U_k \text{Matriz}$

```

     $x \leftarrow Uk(i)$ 
     $UkMatriz(i, x) \leftarrow 1$ 
Fin desde
Dimensionar Ukvector (1,Transiciones) //Vector de 0
Dimensionar Md (Uk,Lugares) //Matriz de 0
Dimensionar Mdfinal(cont,Lugares)
Dimensionar Ukfinal(cont,Transiciones)
Desde i=1 hasta Uk
     $Ukvector \leftarrow UkMatriz(i) + Ukvector$ 
     $Md(i) \leftarrow Mo + (C) * (Ukvector)$  //Expresión (2,1)
Si Md(i) igual que Vacio //Todos sus componentes son 0
     $Mdfinal(cont) \leftarrow Md(i)$ 
     $Ukfinal(cont) \leftarrow UkMatriz(i)$ 
    Si Mdfinal(cont) igual que Mfinal
        Break
    Fin Si
     $cont \leftarrow cont + 1$ 
De lo contrario
     $Ukvector \leftarrow Ukvector - UkMatriz(i)$ 
Fin Si
Fin desde
 $FunObj \leftarrow SumaUkfinal$  //Suma de todos los elementos de la matriz Ukfinal

```

9.2. Identificación de rutas a partir de la matriz de incidencia

Una vez se obtiene la optimización de las redes de Petri por medio de los disparos, surge la necesidad de identificar las diferentes rutas individuales de las RdP, ya que la optimización por disparos es eficaz sólo si no se presenta divergencia y/o convergencia en las redes.

El algoritmo de identificación de rutas se establece en los siguientes pasos:

1. A través de las columnas de la matriz de incidencia se identifican los valores que sean igual a (-1) que representan los arcos que van desde un lugar hacia una transición. Una vez se logra identificar este valor se procede al paso 2. En caso de que exista más de un valor de (-1) indica que hay varias rutas independientes.

2. A través de las filas de la matriz de incidencia se identifican los valores que sean igual a (1) que representan los arcos que van desde una transición hacia un lugar. En caso de que exista más de un valor de (1) indica que hay varias rutas dependientes, es decir, se presentan divergencias o convergencias en la red.
3. Se repiten los pasos anteriores sucesivamente hasta que llegar a la última fila (último lugar). Una vez alcanzado este punto se almacena la ruta establecida con la información de las transiciones en dicho camino y su tiempo asociado.
4. Se repiten los pasos anteriores hasta que se encuentran todas las rutas posibles.

Para identificar las posibles rutas en una red de petri se toma como ejemplo la red de la figura 9.2 se presenta el ejemplo.

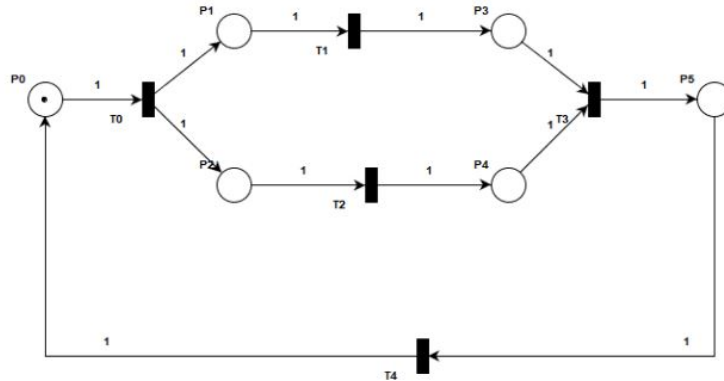


Figura 9.2: Ejemplo identificación de rutas

En la tabla 9.1 se puede ver la matriz de incidencia de la red anterior; y los tiempos asociados a cada transición se presentan en la tabla 9.2

Cuadro 9.1: Matriz de incidencia de figura 9.2

	T0	T1	T2	T3	T4
P0	-1	0	0	0	1
P1	1	-1	0	0	0
P2	1	0	-1	0	0
P3	0	1	0	-1	0
P4	0	0	1	-1	0
P5	0	0	0	1	-1

Cuadro 9.2: Tiempos asociados a las transiciones de la figura 9.2

Transiciones	T0	T1	T2	T3	T4
Tiempos	2	2	3	4	1

Se puede apreciar de la figura 9.2 que la red presenta dos posibles rutas, donde $T0$ es un elemento divergente y $T3$ un elemento convergente. Por lo tanto, es importante hacer un seguimiento de la matriz de incidencia para poder identificar ambas rutas. Las tablas 9.3 y 9.4 muestran los diferentes caminos.

Cuadro 9.3: Ruta 1 descrita de forma matricial.

	T0	T1	T2	T3	T4
P0	-1	0	0	0	1
P1	1	-1	0	0	0
P2	1	0	-1	0	0
P3	0	1	0	-1	0
P4	0	0	1	-1	0
P5	0	0	0	1	-1

En la tabla 9.3 se puede ver que la ruta esta establecida por las transiciones $T0$, $T1$, $T3$ y $T4$, y el tiempo total de esta ruta es de 9 unidades de tiempo.

Cuadro 9.4: Ruta 2 descrita de forma matricial.

	T0	T1	T2	T3	T4
P0	-1	0	0	0	1
P1	1	-1	0	0	0
P2	1	0	-1	0	0
P3	0	1	0	-1	0
P4	0	0	1	-1	0
P5	0	0	0	1	-1

En la tabla 9.4 se puede ver que la ruta esta establecida por las transiciones $T0$, $T2$, $T3$ y $T4$, y el tiempo total de esta ruta es de 10 unidades de tiempo.

Del ejemplo anterior se observa que hay dos rutas, sin embargo ambas son necesarias par generar una secuencia de disparos válida en la red presente. Lo que se desea con este algoritmo es que las rutas sean plenamente identificadas individualmente con sus respectivos tiempos.

Pseudocódigo de la identificación de rutas

Inicio

cargar C // Matriz de incidencia

cargar Tiempo // Vector de tiempos asociados a transiciones

inicializar variables

generar matriz A a partir de C

encontrar 1 de cada fila y columna

encontrar -1 de cada fila y columna

Desde n=1 hasta 100 veces número de filas de C

Desde j=1 hasta columnas de C

Si C(i,j) **menor que** 0 // i contador de filas y j contador de columnas

$A(i, j) \leftarrow C(i, j)$

guardar tiempo de la transición j

Fin Si

Fin Desde

Desde i=1 hasta filas de C

Si C(i,j) **mayor que** 0 // i contador de filas y j contador de columnas

$A(i, j) \leftarrow C(i, j)$

Fin Si

Fin Desde

Si última fila de A **diferente de** 0

$MatA\{h\} \leftarrow A$

$SumTiempo\{h\} \leftarrow sumadetiemposdetransicionesdisparadas$

$VecTiempo\{h\} \leftarrow vectordetransicionesdisparadas$

reiniciar matriz A

reiniciar vector de tiempos

Fin Si

Si MatA{h} se repite //

borrar MatA{end}

borrar SumTiempo{end}

borrar VecTiempo{end}

break

De lo contrario

continuar

Fin Si

Fin Desde

guardar MatA, SumTiempo, VecTiempo

Capítulo 10

Integración entre optimización de disparos y algoritmo de identificación de rutas

Esta integración es capaz de utilizar las fortalezas de las RdP y los AG, para solucionar el problema de la programación en los SMF. El programa implementa un AG que utiliza estructuras de una RdP modelada a partir de un SMF para obtener un tiempo de operación óptimo y una secuencia de disparos asociada a estos resultados.

El AG a partir de un marcado inicial $M0$, codifica una secuencia de disparos de transiciones por medio de los diferentes métodos de evolución y pasos establecidos en la sección 9.1 hasta alcanzar un marcado final deseado (criterio de parada).

Lo que se desea al integrar los dos algoritmos es obtener una secuencia de disparos en función del tiempo, es decir, que la función objetivo sea orientada a identificar la ruta que posea el menor tiempo independientemente de cuantas transiciones fueron sensibilizadas. Es importante resaltar que los tiempos estipulados en este documento son los tiempos de operación de la red y se ven reflejados en los conjuntos lugar-transición como se observa en la figura 10.1.

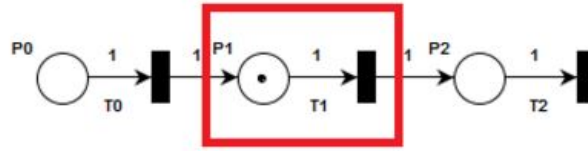


Figura 10.1: Conjunto equivalente al tiempo de operación

Para describir este procedimiento de la integración se toma como ejemplo la red de la figura 9.2 (sección 9.2). Los tiempos de cada evento o conjunto se ven reflejados en la tabla 9.2.

Primero se realiza el procedimiento descrito en la sección 9.2 y se obtienen las rutas individuales y los tiempos respectivos. Para resumir lo obtenido, se presenta la siguiente tabla:

Cuadro 10.1: Identificación de rutas

Rutas	Transiciones	Tiempo
1	T0-T1-T3-T4	9
2	T0-T2-T3-T4	10

Una vez obtenidas las rutas y sus tiempos respectivos, se implementa el algoritmo descrito en la sección 9.1, donde el programa busca el individuo (vector de disparos Uk) más óptimo, pero esta vez su enfoque no está centrado en la menor cantidad de disparos sino en encontrar una secuencia de disparos válida y que el tiempo de operación sea el menor.

Pseudocódigo de Función Objetivo orientada a minimizar tiempo

```

cargar MatA, SumTiempo, VecTiempo, UkMatriz
Dimensionar CopiaUkVector(1,UkVector) //Vector de 0
Desde n=1 hasta MatA // Hasta el número de rutas almacenadas en MatA
    Si UkVector igual VecTiempo(n)
         $TiempoTotal \leftarrow SumTiempo(n)$ 
        break
    Fin Si
Fin Desde
Sí UkVector diferente VecTiempo //Diferente de todos los vectores VecTiempo
    Desde n=1 hasta MatA
        Dimensionar CopiaVecTiempo (1,VecTiempo(n)) //Vector de 0

```

```
Desde n1=1 hasta VecTiempo(n)
Desde n2=1 hasta UkVector
Si VecTiempo(n)(1,n1) igual UkVector(1,n1)
    CopiaVecTiempo(1, n1)  $\leftarrow$  VecTiempo(n)(1, n1)
Fin Si
Si CopiaVector igual VecTiempo(n)
    CopiaUkVector(1, n2)  $\leftarrow$  UkVector(1, n2)
    TiempoTotal  $\leftarrow$  SumTiempo(n)
Fin Si
Fin Desde
Fin Desde
Si CopiaUkVector igual Ukvector
    break
Fin Si
Fin Desde
FunObj  $\leftarrow$  TiempoTotal
Final
```

Capítulo 11

Resultados

En este capítulo se presentan los resultados obtenidos al implementar los diseños de trabajo en el capítulo 8.1 y la metodología del capítulo 10. Cabe resaltar que para poder visualizar los tiempos, estos se deben organizar conforme número de transiciones existan en la RdP. Se evalúan tiempos diferentes en las operaciones con el fin de encontrar la secuencia de disparos óptima y corroborar su correcto funcionamiento.

Todas las RdP que se discuten acá son consideradas como RdP binarias, con un lugar de inicio y un lugar de finalización sin importar que la red sea cíclica o no.

En las figuras de las simulaciones se muestran los resultados asociados a los diferentes casos propuestos. *Mfinal* hace referencia al conjunto de vectores que representan el estado de la red, *UkVector* es el vector de las transiciones disparadas y *fval* es la suma de los tiempos de las transiciones disparadas, es decir, el tiempo que se demora el proceso. En azul, se presenta la la secuencia de disparos obtenida vista desde la RdP. El lugar de inicio (P0 en general), siempre tiene marca, ya que esto indica que siempre hay recursos disponibles.

Se discuten cinco casos para mostrar la funcionalidad del programa donde tres de las redes a trabajar se mencionan en la sección 8.3 y las otras dos redes son topologías especiales que demuestran facultades del programa.

Los parámetros del algoritmo genético que se utilizan para todos los casos se resumen en: selección por torneo, mutación gaussiana, factor de recombinación de 0.5, 50 generaciones y tamaño de población que abarca 200 individuos.

Caso 1. RdP del SMF planta de procesamiento de pan de molde (figura 8.9)

La red de Petri a trabajar se muestra en la figura 8.9. Para realizar un correcto funcionamiento del programa se deben de ingresar previamente la matriz de incidencia como se muestra en la tabla 11.1, vector de estado inicial $M0$ y los tiempos establecidos para cada transición como se muestran en las tablas 11.1 y 11.2 respectivamente.

Cuadro 11.1: Matriz de incidencia caso 1

	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
P0	-1	0	0	0	0	0	0	0	0	0	0
P1	1	-1	0	0	0	0	0	0	0	0	0
P2	0	1	-1	0	0	0	0	0	0	0	0
P3	0	0	1	-1	0	0	0	0	0	0	0
P4	0	0	1	0	-1	0	0	0	0	0	0
P5	0	0	1	0	0	-1	0	0	0	0	0
P6	0	0	0	1	0	0	-1	0	0	0	0
P7	0	0	0	0	1	0	0	-1	0	0	0
P8	0	0	0	0	0	1	0	0	-1	0	0
P9	0	0	0	0	0	0	1	1	0	-1	0
P10	0	0	0	0	0	0	0	1	1	0	-1
P11	0	0	0	0	0	0	0	0	0	1	1

Cuadro 11.2: Vector de estado inicial $M0$ para el caso 1

	P0	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11
Mo	1	0	0	0	0	0	0	0	0	0	0	0

Cuadro 11.3: Tiempos de operación caso 1

Tiempos de operación	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
A	1	2	3	4	5	6	7	8	9	10	11
B	5	5	1	7	6	7	3	3	3	20	25

Con estos datos el programa puede calcular las rutas posibles, el tiempo total de cada una de ellas y asignar un criterio de parada (último lugar de la RdP), además de arrojar como resultado un conjunto de vectores de estados desde $M0$ hasta $Mfinal$, un vector Uk el cual muestra la secuencia de disparos y el valor de tiempo total de operación. La tabla 11.4 contiene las posibles secuencias de disparo, esto es con la finalidad de facilitar la comprensión de los resultados.

Cuadro 11.4: Secuencias de disparos posibles para la Rdp del caso 1

Secuencia de disparos válida	Transiciones disparadas	Suma de tiempos para A	Suma de tiempos para B
Secuencia 1	T0-T1-T2-T5-T8-T10	32	46
Secuencia 2	T0-T1-T2-T4-T7-T10	30	45
Secuencia 3	T0-T1-T2-T4-T7-T9	29	40
Secuencia 4	T0-T1-T2-T3-T6-T9	27	41

Los resultados obtenidos para los tiempos mostrados en la tabla 11.3 se observan a continuación:

Cuadro 11.5: Conjunto de vectores de estados para los tiempos de operación A caso 1

	P0	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11
M0	1	0	0	0	0	0	0	0	0	0	0	0
M1	1	1	0	0	0	0	0	0	0	0	0	0
M2	1	0	1	0	0	0	0	0	0	0	0	0
M3	1	0	0	1	1	1	0	0	0	0	0	0
M4	1	0	0	0	1	1	1	0	0	0	0	0
M5	1	0	0	0	1	1	0	0	0	1	0	0
M6	1	0	0	0	1	1	0	0	0	0	0	1

Cuadro 11.6: Transiciones disparadas para tiempos de operación A caso 1

UkVector	1	2	3	4	7	10
Transición disparada	T0	T1	T2	T3	T6	T9

Por lo tanto, comparando el vector de disparos obtenido con la tabla 11.4, el tiempo total de operación para el suceso A es de 27 unidades de tiempo. En el caso de B se obtuvieron los siguientes resultados:

Cuadro 11.7: Conjunto de vectores de estados para los tiempos de operación B caso 1

	P0	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11
M0	1	0	0	0	0	0	0	0	0	0	0	0
M1	1	1	0	0	0	0	0	0	0	0	0	0
M2	1	0	1	0	0	0	0	0	0	0	0	0
M3	1	0	0	1	1	1	0	0	0	0	0	0
M4	1	0	0	1	0	1	0	1	0	0	0	0
M5	1	0	0	1	0	1	0	0	0	1	1	0
M6	1	0	0	1	0	1	0	0	0	0	1	1

Cuadro 11.8: Transiciones disparadas para tiempos de operación B caso 1

UkVector	1	2	3	5	8	10
Transición disparada	T0	T1	T2	T4	T7	T9

Comparando el vector de disparos obtenido con la tabla 11.4, el tiempo total de operación para B es de 40 unidades de tiempo.

Se puede afirmar entonces que las figuras 11.1 y 11.2 muestran las secuencias de disparo óptimas aplicadas a la RdP para A y B respectivamente.

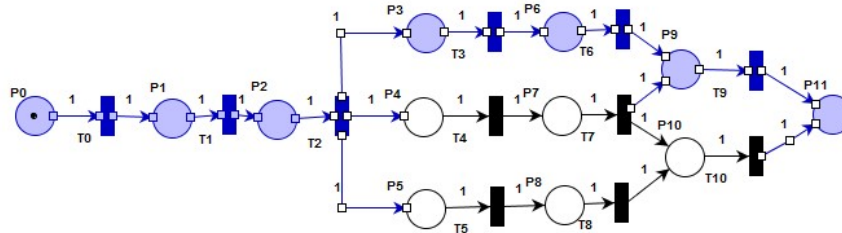


Figura 11.1: Caso 1 resultados vector de tiempo A

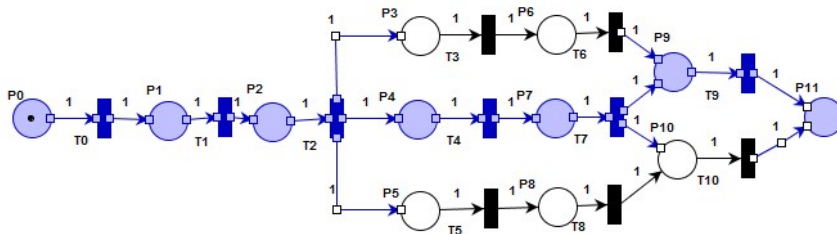


Figura 11.2: Caso 1 resultados vector de tiempo B

Teniendo claro cómo se deben de ingresar los datos y el uso del programa, los casos que se presentan a continuación se muestran solamente los resultados obtenidos y su correspondiente análisis.

Caso 2. RdP del SMF con brazos robóticos (figura 8.10)

Para este caso se utiliza la RdP de la figura 8.10, donde se identifican tres secuencias de disparos posibles que se observan en la tabla 11.9.

Cuadro 11.9: Secuencias de disparos posibles para la RdP del caso 2

Secuencia de disparos válida	Transiciones disparadas	Suma de tiempos para A	Suma de tiempos para B
Secuencia 1	T1-T3-T5-T9-T11-T13	29	42
Secuencia 2	T0-T2-T4-T8-T10-T12	9	48
Secuencia 3	T0-T2-T4-T6-T7-T12	15	53

En la tabla 11.10 se pueden ver los tiempos de operación para el caso 2.

Cuadro 11.10: Tiempos de operación caso 2

Tiempos de operación	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13
A	2	4	1	4	3	6	5	3	1	5	1	3	1	7
B	14	13	12	11	10	9	8	7	6	5	4	3	2	1

Realizando el procedimiento descrito en el caso anterior con los tiempos de operación de la tabla 11.10 se obtienen como resultados los datos de las tablas 11.11 y 11.12 para A.

Cuadro 11.11: Conjunto de vectores de estados para los tiempos de operación A caso 2

	P0	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15
M0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0
M1	1	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0
M2	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0
M3	1	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0
M4	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
M5	1	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0
M6	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1

Cuadro 11.12: Transiciones disparadas para tiempos de operación A caso 2

UkVector	1	3	5	9	11	13
Transición disparada	T0	T2	T4	T8	T10	T12

Por lo tanto, comparando el vector de disparos obtenido con la tabla 11.9, la secuencia 2 es la óptima y el tiempo total de operación para el suceso A es de 9 unidades de tiempo.

El conjunto de vectores de estados y la secuencia de disparos para B se pueden observar en las tablas 11.13 y 11.14 respectivamente.

Cuadro 11.13: Conjunto de vectores de estados para los tiempos de operación B caso 2

	P0	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15
M0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0
M1	1	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0
M2	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0
M3	1	0	0	0	1	0	0	1	0	0	0	1	0	0	0	0
M4	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0
M5	1	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0
M6	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1

Cuadro 11.14: Transiciones disparadas para tiempos de operación B caso 2

UkVector	2	4	6	10	12	14
Transición disparada	T1	T3	T5	T9	T11	T13

La secuencia de disparos óptima para B es la **Secuencia 1**, con un tiempo total de operación de 42 unidades de tiempo.

En ambos sucesos se comprueba que el programa tiende a escoger la secuencia de menor tiempo de operación. Las figuras 11.3 y 11.4 muestran las transiciones disparadas en la RdP para A y B respectivamente.

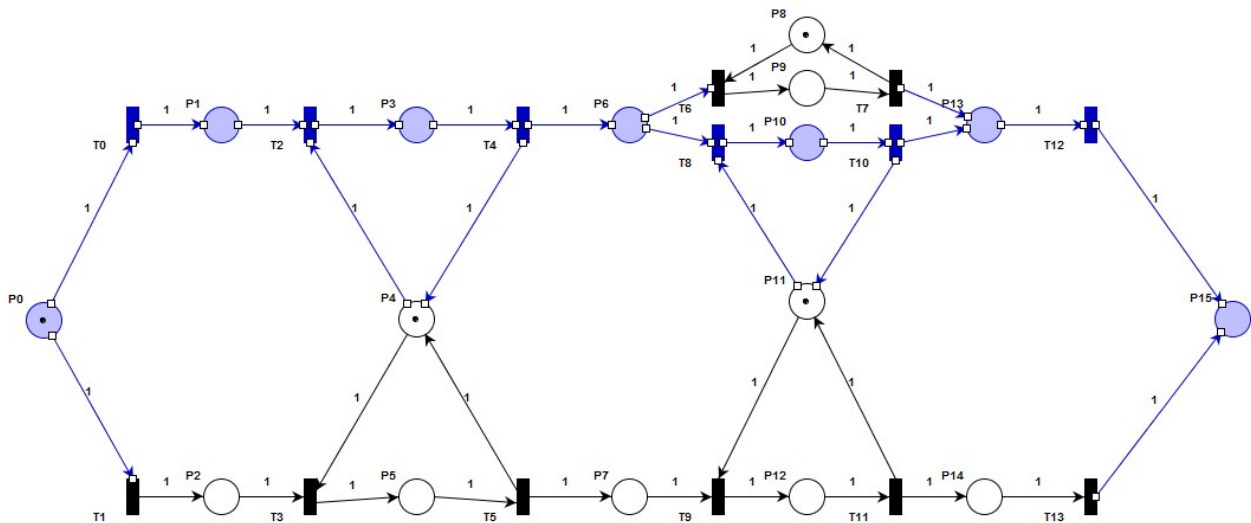


Figura 11.3: Caso 2 resultados vector de tiempo A

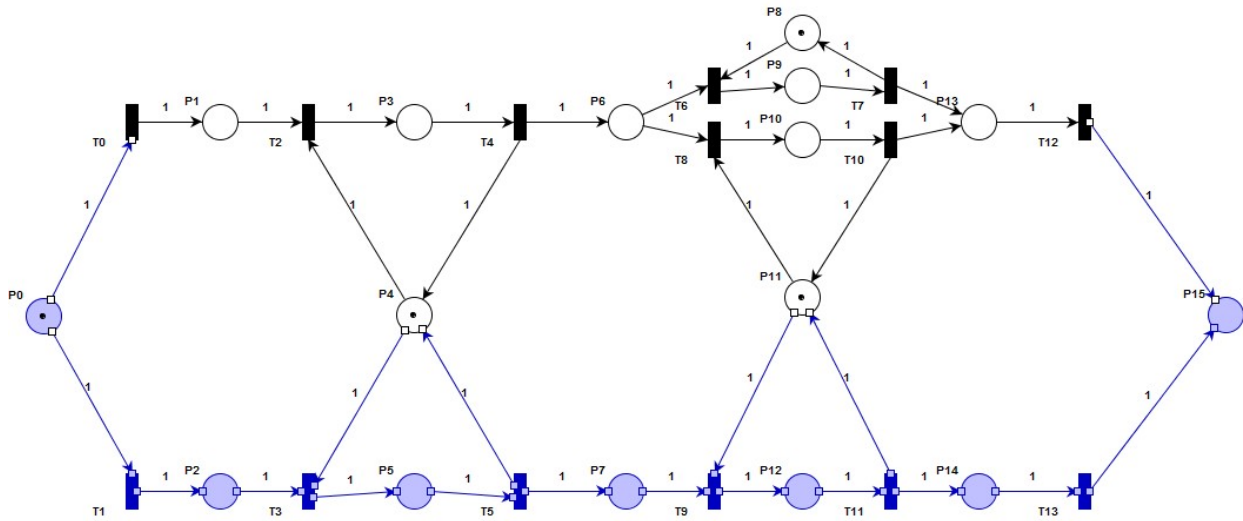


Figura 11.4: Caso 2 resultados vector de tiempo B

Caso 3. RdP del SMF tipo escalera (figura 8.7)

Para este caso se utiliza la red de Petri descrita en la figura 8.7, las posibles secuencias de disparos se encuentran en la tabla 11.15. La tabla 11.16 especifica los tiempos de operación utilizados para este caso.

Cuadro 11.15: Secuencias de disparos posibles para la RdP del caso 3

Secuencia de disparos válida	Transiciones disparadas	Suma de tiempos para A	Suma de tiempos para B
Secuencia 1	T2-T5-T7-T9	20	8
Secuencia 2	T2-T5-T6-T8	22	5
Secuencia 3	T1-T4-T7-T9	15	10
Secuencia 4	T1-T4-T6-T8	17	7
Secuencia 5	T0-T3-T7-T9	19	9
Secuencia 6	T0-T3-T6-T8	21	6

Cuadro 11.16: Tiempos de operación caso 3

Tiempos de operación	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9
A	9	3	4	4	6	10	1	3	7	3
B	1	1	1	3	4	2	1	1	1	4

Si siguiendo el procedimiento descrito en el **caso 1** se obtiene como resultado para el suceso A, el conjunto de vectores de estados en la tabla 11.17 y su correspondiente secuencia de

disparos en la tabla 11.18.

Cuadro 11.17: Conjunto de vectores de estados para los tiempos de operación A caso 3

	P0	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
M0	1	1	0	0	1	0	1	0	1	0	0	1	0
M1	1	1	0	1	0	0	1	0	1	0	0	1	0
M2	1	1	0	0	1	0	1	1	1	0	0	1	0
M3	1	1	0	0	1	0	1	0	1	0	1	0	0
M4	1	1	0	0	1	0	1	0	1	0	0	1	1

Cuadro 11.18: Transiciones disparadas para tiempos de operación A caso 3

UkVector	2	5	8	10
Transición disparada	T1	T4	T7	T9

Se puede afirmar entonces que la secuencia de disparos óptima para el suceso A es descrita por la **secuencia 3** de la tabla 11.15 con un tiempo total de operación de 15 unidades de tiempo.

Los resultados obtenidos con los datos de tiempo del suceso B se muestran en las tablas 11.19 y 11.20 respectivamente.

Cuadro 11.19: Conjunto de vectores de estados para los tiempos de operación B caso 3

	P0	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
M0	1	1	0	0	1	0	1	0	1	0	0	1	0
M1	1	1	0	0	1	1	0	0	1	0	0	1	0
M2	1	1	0	0	1	0	1	1	1	0	0	1	0
M3	1	1	0	0	1	1	0	1	1	0	0	1	0
M4	1	1	0	0	1	1	0	0	0	1	0	1	0
M5	1	1	0	0	1	1	0	0	1	0	0	1	1

Cuadro 11.20: Transiciones disparadas para tiempos de operación B caso 3

UkVector	3	6	7	9
Transición disparada	T2	T5	T6	T8

Para el suceso B se tiene una secuencia de disparos igual a la **secuencia 2** de la tabla 11.15 arrojando como valor óptimo 5 unidades de tiempo.

En las figuras 11.5 y 11.6 se puede observar la secuencia de disparos para el suceso A y B respectivamente.

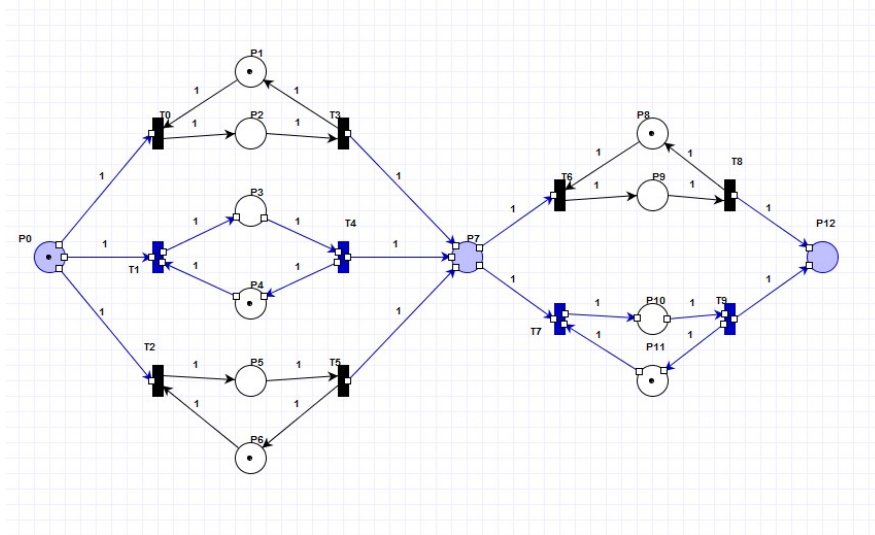


Figura 11.5: Caso 3 resultados vector de tiempo A

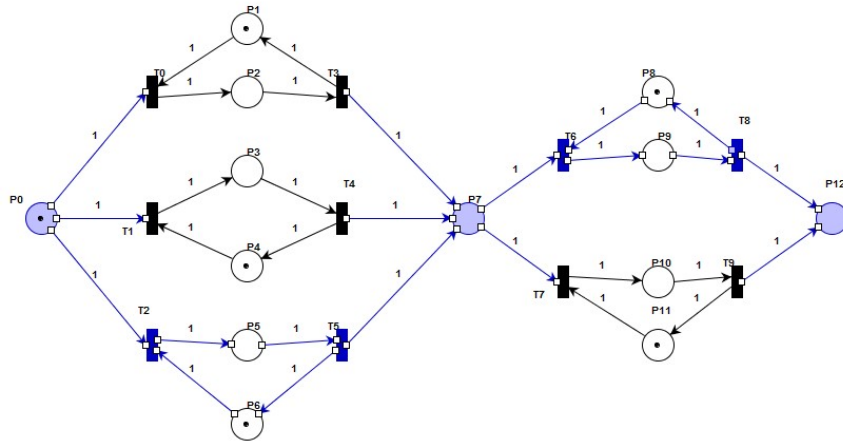


Figura 11.6: Caso 3 resultados vector de tiempo B

Para esta red, con los tiempos especificados se muestran dos de los seis posibles caminos. En este diseño se puede apreciar una buena flexibilidad, ya que dependiendo de las posibles combinaciones en los tiempos se pueden obtener diferentes productos.

Caso 4. RdP de propuesta 1

La red de Petri de la figura 11.7 que se trabaja en este caso presenta la particularidad de tener una secuencia de disparos independiente y una secuencia de disparos dependientes tal y como se menciona en la sección 9.2. Estas secuencias se encuentran descritas por la tabla 11.21. La tabla 11.22 describe los tiempo de operación implementados en este caso.

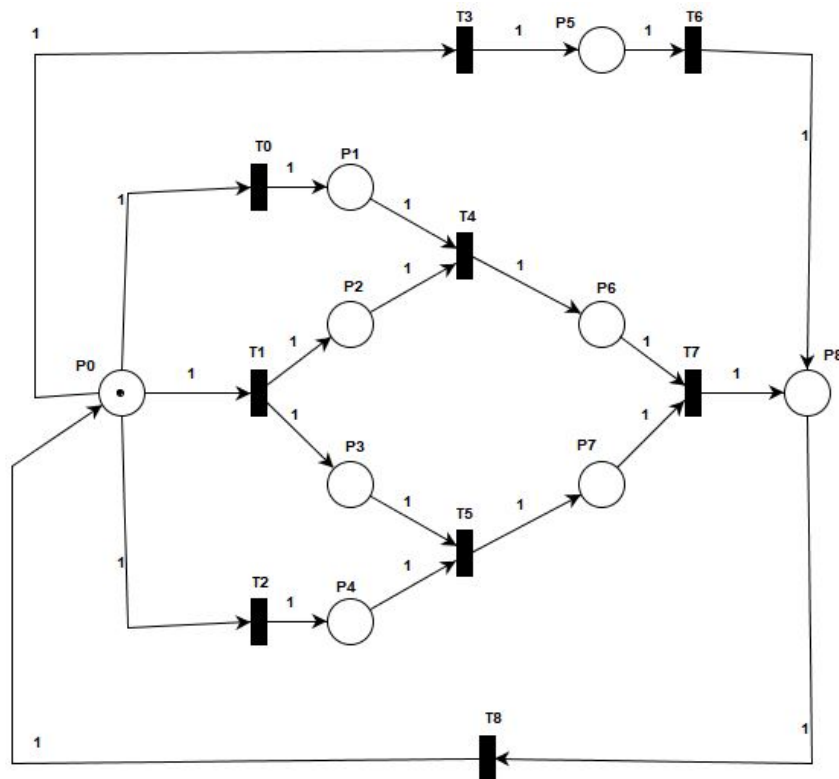


Figura 11.7: Red propuesta para el caso 4

Cuadro 11.21: Secuencias de disparos posibles para la RdP del caso 4

Secuencia de disparos válida	Transiciones disparadas	Suma de tiempos para A	Suma de tiempos para B
Secuencia 1	T3-T6	2	28
Secuencia 2	T0-T1-T2-T4-T5-T7	3	27

Cuadro 11.22: Tiempos de operación caso 4

Tiempos de operación	T0	T1	T2	T3	T4	T5	T6	T7	T8
A	1	1	1	1	1	1	1	1	1
B	10	7	3	13	2	5	15	15	10

La tabla 11.23 representa el conjunto de rutas necesarias para generar la **secuencia 2** de la tabla 11.21. Dado que cada ruta posee un tiempo de operación diferente se debe de tomar el mayor tiempo de las rutas que dependen de esta secuencia, ya que este es el tiempo real de operación.

Cuadro 11.23: Rutas dependientes de secuencia 2

Ruta	Transiciones	Tiempo total suceso A	Tiempo total suceso B
1	T0-T4-T7	3	23
2	T1-T4-T7	3	27
3	T1-T5-T7	3	24
4	T2-T5-T7	3	27

Los resultados obtenidos del suceso A se describen en las tablas 11.24 y 11.25 y los resultados obtenidos del suceso B se describen en las tablas 11.26 y 11.27.

Cuadro 11.24: Conjunto de vectores de estados para los tiempos de operación A caso 4

	P0	P1	P2	P3	P4	P5	P6	P7	P8
M0	1	0	0	0	0	0	0	0	0
M1	1	0	0	0	0	1	0	0	0
M2	1	0	0	0	0	0	0	0	1

Cuadro 11.25: Transiciones disparadas para tiempos de operación A caso 4

UkVector	4	7
Transición disparada	T3	T6

Cuadro 11.26: Conjunto de vectores de estados para los tiempos de operación A caso 4

	P0	P1	P2	P3	P4	P5	P6	P7	P8
M0	1	0	0	0	0	0	0	0	0
M1	1	0	1	1	0	0	0	0	0
M2	1	0	1	1	1	0	0	0	0
M3	1	1	1	1	1	0	0	0	0
M4	1	0	0	1	1	0	1	0	0
M5	1	0	0	0	0	0	1	1	0
M6	1	0	0	0	1	0	1	1	0
M7	1	0	0	0	1	0	0	0	1

Cuadro 11.27: Transiciones disparadas para tiempos de operación B caso 4

UkVector	1	2	3	5	6	8
Transición disparada	T0	T1	T2	T4	T5	T7

Los valores óptimos de tiempo para los sucesos A y B son de 2 y 27 unidades de tiempo respectivamente, donde se aprecia la versatilidad del algoritmo al poder identificar el tiempo óptimo de rutas dependientes. Las figuras 11.8 y 11.9 muestran la secuencia de disparos sobre la RdP para ambos sucesos.

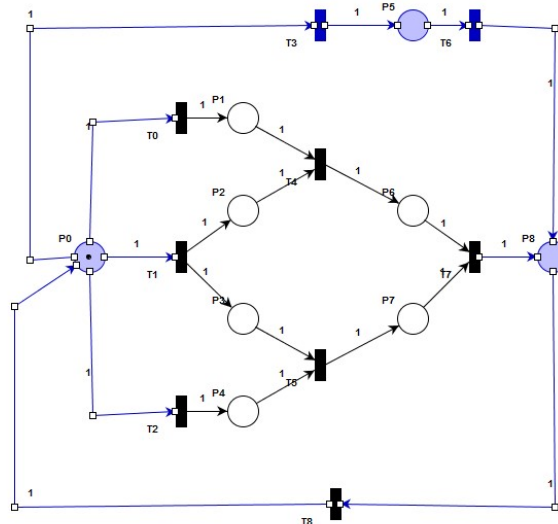


Figura 11.8: Caso 4 Resultados Vector de tiempo A

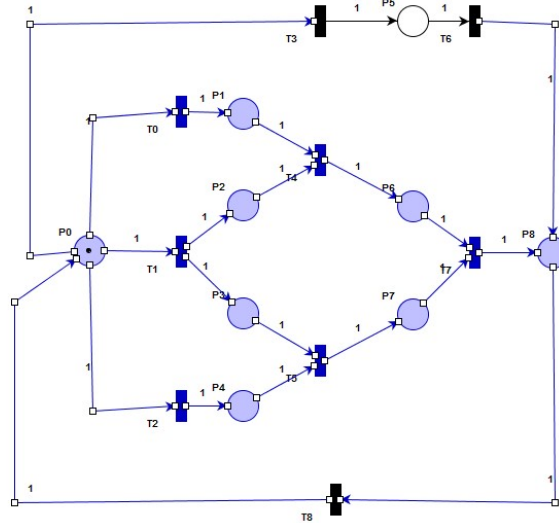


Figura 11.9: Caso 4 Resultados Vector de tiempo B

Caso 5. RdP propuesta 2

La red de Petri de la figura 11.10 tiene la particularidad de poseer una gran flexibilidad al ofrecer 36 secuencias de disparo posibles, esto se puede interpretar como un SMF de gran flexibilidad a la hora de procesar un producto. La tabla 11.28 describe los tiempos de operación implementados para este caso.

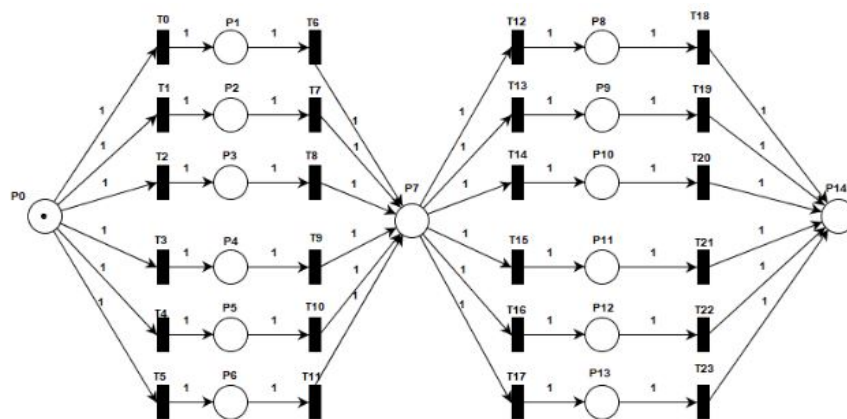


Figura 11.10: Red propuesta para el caso 5

Cuadro 11.28: Tiempos de operación caso 5

Tiempos de operación	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11
A	9	3	2	8	9	2	8	1	3	9	3	7
B	20	20	2	18	9	3	12	10	4	1	13	18
Tiempos de operación	T12	T13	T14	T15	T16	T17	T18	T19	T20	T21	T22	T23
A	1	9	6	4	7	4	3	1	9	4	5	3
B	6	5	1	15	17	15	10	17	9	9	1	13

El conjunto de vectores de estados y la secuencia de disparos resultantes del suceso A se encuentran en las tablas 11.29 y 11.30, el tiempo de operación óptimo es de 8 unidades de tiempo.

Cuadro 11.29: Conjunto de vectores de estados para los tiempos de operación A caso 5

	P0	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14
M0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
M2	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0
M3	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
M4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Cuadro 11.30: Transiciones disparadas para tiempos de operación A caso 5

UkVector	2	8	13	19
Transición disparada	T1	T7	T12	T18

Para el suceso B se tiene un conjunto de vectores de estados descrito en la tabla 11.31 y la secuencia de disparos descrita por la tabla 11.32, el tiempo de operación óptimo es de 16 unidades de tiempo.

Cuadro 11.31: Conjunto de vectores de estados para los tiempos de operación B caso 5

	P0	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14
M0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
M2	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0
M3	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0
M4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Cuadro 11.32: Transiciones disparadas para tiempos de operación B caso 5

UkVector	3	9	15	21
Transición disparada	T2	T8	T14	T20

En esta topología dado el tiempo óptimo se verifica con la secuencia de disparos Uk que sin importar el número de rutas o elementos el algoritmo tiende a encontrar el valor óptimo.

Implementando diferentes topologías que hacen referencia a diferentes tipos de SMF se comprueba el correcto funcionamiento del programa sin importar la diversidad de las redes.

Capítulo 12

Conclusiones

- Según lo propuesto en diferentes documentos de la literatura, se estableció una metodología de modelado de Sistemas de Manufactura Flexible por medio de Redes de Petri, donde las topologías básicas de los SMF y dicha metodología se combinaron para dar como resultado una RdP capaz de satisfacer las demandas operacionales de manera eficaz. Además se indagó sobre las teorías de los Algoritmos Genéticos como herramienta de optimización y se determinó su utilidad frente a la problemática que albergan los SMF, puesto que logran minimizar los tiempos de operación sin importar su complejidad.
- Se determinó una metodología para modelar SMF mediante RdP capaz de reflejar el funcionamiento del sistema deseado. Partiendo desde el diseño y aplicando teorías de prevención y evitación de puntos muertos con el fin de obtener el mejor desempeño de los recursos del sistema.
- De acuerdo a las metodologías de optimización investigadas, se implementó una secuencia de pasos capaz de adaptar las problemáticas encontradas en los SMF modelados a través de RdP cuya función objetivo es la de encontrar solución a la programación de operaciones y tiempos de ejecución de procesos.
- Se estableció un algoritmo identificador de rutas a partir de la matriz de incidencia de las Redes de Petri, el cual se encarga de identificar los diferentes caminos presentes en una red, detectar las transiciones disparadas en estos y guardar su tiempo de ejecución. Con el fin de formar un lazo entre los SMF y su modelamiento a través de las RdP.
- Implementando modelos encontrados en la literatura y algunos propuestos, se pudo verificar el correcto funcionamiento del programa a través de la comparación de datos teóricos con los resultados obtenidos.

- La metodología de diseño de Redes de Petri está enfocada para la creación de proyectos de Sistemas de Manufactura Flexible que en conjunto con el programa de Algoritmos Genéticos permite evaluar el diseño efectuado de tal forma que se puedan optimizar los recursos a disposición del proyecto. Ofreciendo resultados de calidad ya que permite minimizar problemáticas recurrentes a las capacidades operativas de automatismos enfocados en Sistemas de Manufactura Flexible

Futuros trabajos de investigación

- Para los trabajos se recomienda expandir la capacidad del programa de tal forma que abarque Redes de Petri que no sean 1 limitadas, además de poder visualizar múltiples rutas óptimas dependiendo de la red
- Mejorar o rediseñar el algoritmo a tal punto que no sea necesario empezar y terminar las RdP con un sólo lugar, es decir, que el algoritmo tenga la capacidad de ejecutar procesos con varios lugares de inicio y varios lugares de finalización.

Bibliografía

- [1] C. E. Flores, “Sistema de fabricación flexible: Fms,” *Universidad Rafael Landívar: Boletín electrónico [en línea].*[consultado 20.04. 2017] Disponible en: http://www.fgsalazar.net/LANDIVAR/ING-PRIMERO/boletin03/URL03_MEC02.pdf.
- [2] M. Bashir, B. B. Muhammad, and Z. Li, “Minimal supervisory structure for flexible manufacturing systems using petri nets,” in *Control, Automation and Robotics (ICCAR), 2016 2nd International Conference on.* IEEE, 2016, pp. 291–296.
- [3] C. Rovetto, “Métodos basados en redes de petri para el diseno de algoritmos de encaminamiento adaptativos minimos libres de bloqueos,” Ph.D. dissertation, PhD thesis, University of Zaragoza, Zaragoza, 2011.
- [4] M. C. C. Eduardo, “Algoritmos de enrutamiento.”
- [5] D. González Parra *et al.*, “Optimización por algoritmos genéticos y redes de petri del tiempo de proceso de un sistema de manufactura flexible,” B.S. thesis, Pereira: Universidad Tecnológica de Pereira, 2014.
- [6] G. Mušić and D. Matko, “Petri net based supervisory control of flexible batch plants,” in *Prepr. 8th IFAC Symp. on Large Scale Systems: Theory & Application*, vol. 2, 1998, pp. 989–994.
- [7] M. L. L. Agost and J. O. Villarroja, *Redes reconfigurables: Modelización y verificación*. Universidad Politécnica de Valencia, 2003.
- [8] C. Jaramillo, “Sistemas de manufactura flexible un enfoque estructural.” *Revista Páginas*, no. 16, 1986.
- [9] E. Córdoba Nieto, “Manufactura y automatización,” *Ingeniería e Investigación*, vol. 26, no. 3, pp. 120–128, 2006.
- [10] J. Arranz de la Peña and A. Parra Truyol, “Algoritmos genéticos,” *Recuperado el*, vol. 20, pp. 06–07, 2007.

- [11] P. T. Rodríguez-Piñero, *Introducción a los algoritmos genéticos y sus aplicaciones*. Universidad Rey Juan Carlos, Servicio de Publicaciones, 2003.
- [12] M. P. Groover, *Fundamentals of modern manufacturing: materials processes, and systems*. John Wiley & Sons, 2007.
- [13] H. Machinery, “Cnc robotics & machine tool robotics.” [Online]. Available: <https://www.hillaryinc.com/cnc-robotics.htm>
- [14] INDEVA, “Automatic guided vehicles – indeva® agv.” [Online]. Available: <http://www.indevagroup.com/automatic-guided-vehicles/>
- [15] L. D. Murillo-Soto, “Redes de petri: Modelado e implementación de algoritmos para autómatas programables,” *Revista Tecnología en Marcha*, vol. 21, no. 4, p. 102, 2008.
- [16] M. Holguin, A. Orozco, and C. Guarnizo, “Automatismos industriales,” *Taller de publicaciones Universidad Tecnológica de Pereira*, pp. 5–11, 2008.
- [17] J. P. C. Villalobos and G. Mejía, “Redes de petri y algoritmos genéticos, una propuesta para la programación de sistemas de manufactura flexible,” *Ingeniería y Universidad*, vol. 10, no. 1, 2010.
- [18] M. H. F. Al Hazza, A. H. Taha, E. Y. Adesta, and A. Albakri, “Modelling and analysing deadlock in flexible manufacturing system using untimed petri net,” in *Advanced Computer Science Applications and Technologies (ACSAT), 2015 4th International Conference on*. IEEE, 2015, pp. 261–265.
- [19] P. E. Valencia, “Optimización mediante algoritmos genéticos,” in *Anales del Instituto de Ingenieros de Chile*, vol. 109, no. 2, 1997, pp. 83–92.
- [20] T. Ocampo, E. Mirledy *et al.*, “Solución del problema de empaquetamiento óptimo bidimensional,” Ph.D. dissertation, Universidad Tecnológica de Pereira, 2007.
- [21] A. L. C. López and G. Mejía, “Combinación de métodos heurísticos con redes de petri para la programación de sistemas de manufactura flexible,” Ph.D. dissertation, Uniandes, 2006.